COMPUTING SURFACE UNIFORMIZATION USING DISCRETE BELTRAMI FLOW

Abstract. In this paper, we propose a novel algorithm for computing surface uniformization for surfaces with arbitrary topology. According to the celebrated uniformization theorem, all Riemann surfaces can be classified as elliptic, parabolic or hyperbolic. Our algorithm is able to work on all these cases by first constructing an initial map onto an appropriate domain, such as sphere, or a polygon in the plane $\mathbb{R}^2$ or the hyperbolic disk D, and then morphing the diffeomorphism based on the discrete Beltrami flow algorithm. For high genus surfaces, both the final mapping and the target domain is unknown, which presents a challenge in general. Each such surface can be conformally mapped onto D modulo a discrete subgroup of all fractional linear transforms on D. A conformal copy of the surface, which is also its uniformization domain, can be visualized as a fundamental polygon in D corresponding to the discrete subgroup, where the generators of the discrete subgroup map each side of the fundamental polygon to its corresponding side, giving the conformality information of the surface. The novelty in our method lies in the iterative change of these generators as the diffeomorphism is morphing, which indicates a change of geometry of the target domain to match the geometry of the original surface. Numerical results are presented to show the efficiency and accuracy (in terms of distortion) of our methods as well as comparison to other state of the art algorithms.

Key words. uniformization, discrete Beltrami flow, quasiconformal mapping

AMS subject classifications. 30F10, 30L10, 53A05, 53A30

1. Introduction.

Surface uniformization is an important mathematical tool for geometric study and understanding of surfaces. By mapping surfaces conformally onto a canonical domain, we have a natural correspondence between surfaces of the same topology class. Therefore, it is widely used in applications such as surface processing, texture mapping, medical imaging, shape analysis and surface classification and comparison. Many of these applications rely on surface uniformization as an essential tool for understanding the geometry of surfaces. Therefore, it is important to have efficient computational tools for constructing surface uniformizations in practice.

A number of approaches have been proposed for surface uniformization based on different but equivalent mathematical characterization of surface uniformization. For examples, one class of approaches compute surface uniformizations by solving a system of linear equations. The uniformization problem is first formulated as an equivalent optimization problem, such as the minimization of harmonic energy. Then a linear system is constructed for solving the optimization problem, usually by finite element method. However, such methods can only work on a small class of surfaces, such as simply connected open surfaces where such a reformulation exists. Therefore, they are efficient but do not work for general multiply connected domains with arbitrary topology. For these surfaces, such as those simply connected surfaces, one can use a gradient descent approach to minimize the harmonic energy. However, as we will show, computational methods based on harmonic energy may not be very accurate and hence cause quite large angular distortions in practice.

An alternative approach is based on Ricci flow, which aims to adjust the conformal factor of a surface map such that the surfaces metric evolves according to Ricci flow to flatten the surface. As the surface is flattened to make its Gaussian curvature as close to 0 as possible, the surface map becomes the desired surface uniformization. It is known that Ricci flow is the gradient flow that minimizes a convex energy for Euclidean and hyperbolic domains. With the energy being convex and smooth, Newton's method can be effectively used to minimize the energy, and it can be shown that Ricci flow methods converge exponentially fast. However, on spherical domains, the energy is non-convex and singularities may occur, which have to be removed by some surface surgeries. Despite the fact, Ricci flow based methods are considered very efficient and work for surfaces with general topologies.

In this paper we propose a new approach based on quasiconformal geometry that allows the use of general energy functionals in terms of the Beltrami differential, which describes non-conformal distortions of surface maps. Our formulation is more general and flexible while simple to understand and implement. A simple least squares energy functional of the Beltrami differential can be used to compute surface uniformizations that result in minimal distortions comparable to that by Ricci flow based methods.

A sketch of our approach is as follows. We first cut open the surface to obtain a topological disk. Then we compute an initial map from M to the parametrization target domain, which is a sphere for genus 0 surfaces, a rectangle for genus 1 surfaces, and a regular polygon in the hyperbolic disk for high genus surfaces. The rectangle and regular polygons are standard fundamental polygons for genus one and high genus surfaces respectively. The generators for their transformation groups can be easily computed by solving for the parameters of the translation or Möbius transform that bring an edge of the fundamental polygon to its corresponding edge. The initial map is computed through Tutte mapping or the discrete conformal embedding. With the initial map, we adjust the parametrization using the discrete Beltrami flow algorithm, which computes a vector field that adjusts the parametrization to minimize the L^2-norm of the Beltrami differential. However, one difficulty for computing uniformization for high genus surface lies at the fact that both the final mapping and the target domain which are coupled needs to be determined by the flow concurrently. In our approach we design a flow that keeps the consistency of corresponding points under the change of generators for high genus surfaces. With this technique, we can adjust the diffeomorphism and the generators, which determine the target domain, at the same time to achieve the correct conformal structure for surfaces with genus $g \geq 1$. This technique is useful for keeping proper conformal structures of high genus surface parametrizations, and can be directly applied to other high genus surface mapping applications, such as surface registrations.

2. Previous Work. Surface uniformization is an essential tool for various applications of surface processing, including brain mapping [8, 11], texture mapping [10, 20, 23], shape analysis [27] and surface classification [9] and comparison [17].

Several different computational methods for surface uniformization were introduced recently, most of which are after 2000 as computational power improved. Lévy et al. [20] proposed to generate texture atlas by computing conformal maps in the least squares sense. The technique can also be applied to surface uniformization for simply connected open surfaces. Gu et al. [8] proposed computing genus zero conformal maps by minimizing the harmonic energy functional using gradient descent. Jin et al. [14] proposed computing global conformal structures of surfaces by computing the holomorphic 1-forms for surfaces of genus $g \geq 1$. Open surfaces are modeled as closed surfaces by a double covering. Later, Jin et al. [13] proposed using discrete surface Ricci flow for the uniformization of surfaces with arbitrary topology. The method was improved upon the work of Chow et al. [3], which connected Ricci flows on discrete surfaces with those on continuous surfaces, but used a slower gradient descent approach. In general, Ricci flow approaches try to flatten the curvature of discrete surfaces by changing the metric, while our approach considers the problem from a mapping perspective.

The approach of this paper is to take another view on the uniformization problem for discrete surfaces. Note that it is not possible to compute uniformizations for discrete surfaces which are exactly conformal, because the angle sum around each vertex may not add up to precisely π. Therefore it is more practical to consider maps with bounded conformal distortions and be able to control the desired amount of distortions. Hence, the theory of quasiconformal maps is closely related to our method. It was a natural extension of the development of conformal maps for discrete surfaces. Using the method of Beltrami holomorphic flow, Lui et al. [19] proposed to represent and adjust surface registrations using Beltrami coefficients. The method uses an integral flow formula and is restricted to simply connected surfaces. By making use of auxiliary metrics, Zeng et al. [26] extended the method of discrete surface Ricci flow to general quasiconformal maps for arbitrary topologies. The computation of quasiconformal maps was studied by Mastin et al. [21], where a weak formulation was used to solve Beltrami equations for special class of domains with slits. Using a linear discretization for the Beltrami equation, Lui et al. [18] proposed an efficient Beltrami representation for applications in texture mapping and video compression. By deriving simple formulas for quasiconformal maps fixing 4 points, Lipman et al. [16] computed plane deformations with minimal distortions and applied the method on the adjustment of planar maps. The idea of quasiconformal maps can also be described as maps with a bounded distortion, and was used by Lipman [15] for finding maps from triangular meshes onto the plane with desirable properties. Recently, Wong and Zhao [25] proposed a discrete Beltrami flow directly between two arbitrary surfaces. The flow is designed in term of a vector field corresponding to the adjustment to the intrinsic Beltrami differential defined in a local conformal coordinate. Using a least squares approach, the method can compute quasiconformal homeomorphisms that minimize energies in terms of Beltrami differentials between two arbitrary surfaces and can incorporate other constraints, such as correspondences and curve/boundary conditions, in the formulation.

3. Theoretical Background.

In this section, we give an overview of the technical background of our surface uniformization algorithms. To work on general surfaces with arbitrary topology, we first introduce surface classification results from the Riemann uniformization theorem. This enables us to work on a canonical domain such as the sphere, the Euclidean plane or the hyperbolic disk. Then we introduce how quasiconformal maps between surfaces can be represented by Beltrami differentials on these domains. This allows us to describe distortions in surface maps. Using the Beltrami differential as our representation, we consider how the change of it is related to the vector field we use to adjust surface maps. Precisely, we introduce the method of Beltrami flows to compute such vector fields. Finally, to work on high genus surfaces, we derive the consistent conditions the Beltrami flows have to satisfy while adjusting the surface maps.

3.1. Closed Surfaces and Their Uniformizations.

The uniformization theorem [1] states that every closed smooth surface can be biholomorphically mapped onto exact one of the unit sphere, the Euclidean plane or the hyperbolic disk, after a quotient operation by a discrete subgroup acting on the uniformizing space for the latter 2 cases. For the case of simply connected closed surfaces, they can always be conformally mapped onto the unit sphere without any quotient operation. The process of finding such conformal maps is called surface uniformization. To precisely define the notations we use for cases requiring quotient operations, we describe the terminology we use and the uniformization process for the genus one and high genus

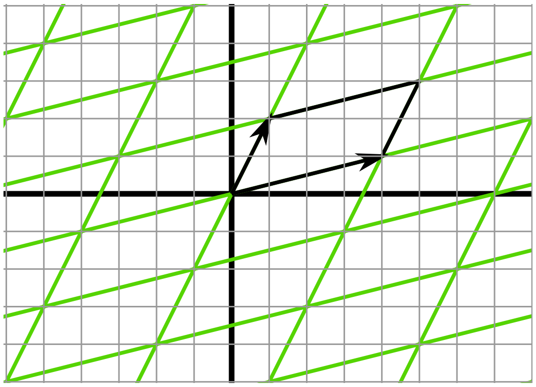


Fig. 3.1. *An example of the fundamental parallelogram for a genus one surface. In this example, the generators* $(4, 1)$ *and* $(1, 2)$ *represent the translations* $T_1(x, y) = (x + 4, y + 1)$ *and* $T_2(x, y) = (x+1, y+2)$ *of the plane, generating the transformation group of the parallelograms tessellating the plane.*

cases in the next 3 subsections.

3.2. The Notions of Transformation Groups and Generators. The uniformizing map of a surface is a conformal map from its universal covering space onto the surface. For genus one and high genus surfaces, this mapping is not one-to-one and satisfy some periodic boundary conditions, and the notions of transformation groups and generators are very useful for understanding this situation. For genus one and high genus surfaces, they can be cut open and mapped onto a copy of itself in the uniformizing space through the uniformizing map. Each such copy of a genus one surface is called the fundamental parallelogram and that of a genus g surface is called the fundamental polygon (with $4g$ sides). The polygons satisfy certain constraints which allow them to tessellate the uniformizing spaces, and some correspondence of their sides are given so that each copy can be "stitched" to form the original surface. In fact, the uniformizing maps are universal coverings of the surfaces. Associated with each universal covering is the transformation group, which consists of automorphisms, or bijective diffeomorphisms, of the uniformizing space that sends a copy of the surface to another. For a genus one surface, this transform group has 2 generators, which are translations of $\mathbb{R}^2$. For a high genus surface with genus g, the transformation group consists of $2g$ generators which can be represented as fractional linear transforms of the hyperbolic disk. Such transformation group is called a Fuchsian group. In this way, copies of the surface tessellate the uniformizing space under the action of its transformation group. Two intuitive figures are shown in Figures 3.1 and 3.2 to demonstrate this. Therefore the generators completely determine the conformal geometry of the surfaces and finding them correctly is essential to our algorithms. We will understand each of the genus one and high genus cases in depth in the following subsections. A discussion of the transformation groups for $\mathbb{R}^2$ can be found in [1]. For an introduction to Fuchsian groups, we refer interested readers to [6].

3.3. Uniformization for Genus One Surfaces. According to the uniformization theorem, every genus one closed surface M can be mapped conformally onto $\mathbb{R}^2/T$, where T is a discrete additive subgroup of $\mathbb{R}^2$ generated by 2 generators. Without loss of generality, we may assume the generators to be $(1, 0)$ and (a, b). M can be mapped conformally onto a fundamental domain Ω with 4 corners identified. Denote

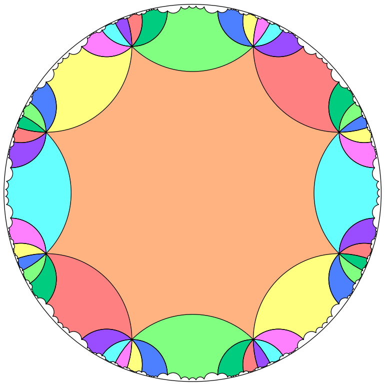


FIG. 3.2. *An example of fundamental octagon for genus two surfaces. In this example, the regular octagon with angles of $\pi/4$ tessellate the hyperbolic disk with a suitable transformation group with 4 generators.*

by p_1, p_2, p_3 and p_4 the corners of the fundamental domain. With the generators, $p_1 = (0,0)$, $p_2 = (1,0)$, $p_3 = (1+a,b)$ and $p_4 = (a,b)$. Let A_1, B_2, A_2 and B_1 be the boundaries of Ω from p_1 to p_2, p_2 to p_3, p_3 to p_4 and p_4 to p_1 respectively. By the generator relations, we have $A_2 = A_1 + a + b\sqrt{-1}$, and $B_2 = B_1 + 1$. Therefore, to reach the correct generators in our algorithms, these relations must be satisfied as Ω is deformed under the Beltrami flow, with respect to the changing generator $(a(t), b(t))$.

3.4. Uniformization for High Genus Surfaces. According to the uniformization theorem, every high genus surface M with genus greater than or equal to 2 admits a uniformization onto the hyperbolic upper half plane $\mathbb{H}$. The set of all biholomorphic self-maps of $\mathbb{H}$ is given by the group $PSL(2,\mathbb{R}) := SL(2,\mathbb{R})/\{\pm I\}$. Consider $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in PSL(2,\mathbb{R})$. It acts on $\mathbb{H}$ by the action

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot z := \frac{az+b}{cz+d} \quad \forall z \in \mathbb{H}. \tag{3.1}$$

With a conjugation, we can think of $PSL(2,\mathbb{R})$ acting on the unit disk D as well, since D and $\mathbb{H}$ are conformally equivalent. Mathematically, there exists a discrete subgroup $G \subset PSL(2,\mathbb{R})$, called the Fuchsian group, such that there is a conformal diffeomorphism $\phi \colon M \to D/G$. Precisely, the Fuchsian group is a discrete subgroup of $PSL(2,\mathbb{R})$, which acts on the hyperbolic disk by fractional linear transformations, also called Möbius transformations.

To correctly adjust a diffeomorphism from a surface onto D/G, one needs to continuously morph the generators of G such that the corresponding boundaries remain consistent. For a genus g surface, it is possible to realize D/G as a fundamental polygon in D with $4g$ sides. Denote by $p_1, p_2, \ldots, p_{4g}$ the vertices of the fundamental polygon. One can find generator $a_1, \ldots, a_g, b_1, \ldots, b_g$ such that $a_1(p_4) = p_1, a_1(p_3) = p_2, b_1(p_2) = p_5, b_1(p_3) = p_4, \ldots, a_g(p_{4g}) = p_{4g-3}, a_g(p_{4g-1}) = p_{4g-2}, b_g(p_{4g-2}) = p_1, b_g(p_{4g-1}) = p_{4g}$. Then the generators automatically satisfy the relation

$$a_1 b_1 a_1^{-1} b_1^{-1} \ldots a_g b_g a_g^{-1} b_g^{-1} = \mathbf{id}. \tag{3.2}$$

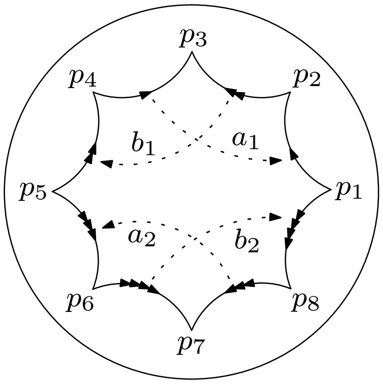


FIG. 3.3. *The fundamental polygon for a genus two surface. Each pair of corresponding sides is labeled with the same number of arrows. The actions of the generators of its Fuchsian group are shown by the dotted arrows.*

Therefore the Fuchsian group G is generated by these generators:

$$G = \langle a_1, \dots, a_g, b_1, \dots, b_g \rangle. \tag{3.3}$$

A figure of the fundamental polygon for a genus two surface is shown in Figure 3.3 with the generators of its Fuchsian group and corresponding sides labeled.

The key issue for constructing the uniformization for high genus surfaces is that both the target domain, i.e., the fundamental polygon, and the conformal map between the surface and the fundamental polygon need to be constructed. That means the generators and hence the fundamental domain have to be adjusted together with the diffeomorphisms between the surface and fundamental domain during the flow. In particular one needs to keep the lengths equal for the $2g$ pairs of corresponding sides and preserve the angle sum of the fundamental domain to 2π. In discussing our algorithms, we will illustrate these constraints in detail and show that they can be enforced by modifying the Beltrami flow. Before explaining our method for this more complicated case, we first describe the geometry of the hyperbolic disk and its transformations in the next subsection.

3.5. Geometry of the Hyperbolic Disk. The hyperbolic disk D, also known as the Poincaré disk, can be constructed by equipping the unit disk in the complex plane with the following metric:

$$ds^2 = \frac{dz\, d\overline{z}}{(1 - z\overline{z})^2}. \tag{3.4}$$

With the above metric, the hyperbolic distance from a point $p \in D$ to the origin is

$$\mathrm{dist_H}(z) = \tan^{-1}(|z|), \tag{3.5}$$

where H denotes hyperbolic. It can be shown that all isometries of the hyperbolic upper half plane $\mathbb{H}$ are of the form

$$\phi(z) = \frac{az + b}{cz + d}, \tag{3.6}$$

where a, b, c, d are real numbers and $ad - bc \neq 0$. Therefore the group $PSL(2,\mathbb{R})$ acts on $\mathbb{H}$ by Möbius transformations. For the hyperbolic disk, the isometries take the form

$$w = e^{\phi\sqrt{-1}} \frac{z-a}{1-\overline{a}z}, \tag{3.7}$$

where $\phi \in \mathbb{R}$ and $a \in D$. This will be the form we use to describe the parameters of our generators. The group of Möbius transformations of this form is isomorphic to $PSL(2,\mathbb{R})$.

3.6. Beltrami Coefficients and Beltrami Differentials. With the theory on surfaces and their uniformization, it is important to have a mathematical tool to describe the distortions of surface maps, which allows us to measure how far away we are from the uniformizing map. For this purpose, we naturally consider the theory of quasiconformal maps, which is a direct generalization of conformal maps. In this subsection, we give a brief introduction on quasiconformal maps, Beltrami coefficients and Beltrami differentials. A detailed introduction on the theory of quasiconformal maps can be found in [2] and [7].

Consider first a complex-valued function f defined on the complex plane. f can also be considered as a complex-valued function defined on the (x,y)-plane $\mathbb{R}^2$. Assume that both real and imaginary parts of f are C^1 as real-valued functions. Its Beltrami coefficient at z is defined to be

$$\mu(z) = \frac{f_{\overline{z}}(z)}{f_z(z)}, \tag{3.8}$$

where $f_{\overline{z}}$ and f_z are complex derivatives of f defined as

$$\frac{\partial f}{\partial \overline{z}} = \frac{1}{2}\left(\frac{\partial f}{\partial x} + \sqrt{-1}\frac{\partial f}{\partial y}\right) \tag{3.9}$$

and

$$\frac{\partial f}{\partial z} = \frac{1}{2}\left(\frac{\partial f}{\partial x} - \sqrt{-1}\frac{\partial f}{\partial y}\right). \tag{3.10}$$

(3.8) is also called the Beltrami equation, and a function f satisfying (3.8) is said to be quasiconformal. Locally f maps a small disk centered at z to an ellipse centered at $f(z)$, with the dilation $K(z)$ given by $\frac{1+|\mu(z)|}{1-|\mu(z)|}$. If there is no non-conformal distortion at z, then $\mu(z) = 0$ and $K(z) = 1$. Therefore the Beltrami coefficient is a quantity that measures local non-conformal distortions of surface maps.

To generalize the notion of Beltrami coefficients to arbitrary Riemann surfaces, where a global conformal parametrization may not exist, we can represent the local non-conformal distortions of a surface map by the Beltrami differential

$$\mu(z)\frac{\overline{dz}}{dz}, \tag{3.11}$$

where $\mu(z)$ is the Beltrami coefficient of the surface map represented using a local conformal parametrization z. In 4.1, we will describe the discrete Beltrami flow algorithm, which allows us to adjust the Beltrami differentials of surface maps using vector fields.

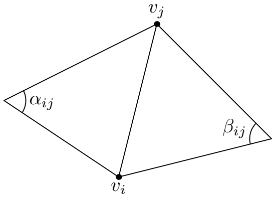


FIG. 3.4. *With adjacent vertices v_i and v_j on the original mesh, the angles α_{ij} and βij are the labeled angles from the adjacent triangles of the ij-th edge.*

3.7. Initial Map. Before starting discrete Beltrami flow to compute the uniformization for a given surface, we need to cut open the closed surface and construct an initial map from the surface onto a planar domain. Here we propose two options.

3.7.1. Tutte Embedding. To map a cut open graph onto a planer domain, we apply the well known Tutte embedding [24], which asserts that for any 3-connected graph $G = \langle V, E, F\rangle$ representing a triangular mesh, suppose its boundary ∂G is embedded in the plane as a (not necessarily strictly) convex polygon, then the following systems of linear equations can be solved for the x and y coordinates of the embedding of G:

$$\sum_{v_j \in N(v_i)} \frac{1}{|N(v_i)|} x_j = x_i, \quad i = 1, \ldots, |V| - |\partial G|, \tag{3.12}$$

$$\sum_{v_j \in N(v_i)} \frac{1}{|N(v_i)|} y_j = y_i, \quad i = 1, \ldots, |V| - |\partial G|, \tag{3.13}$$

where $\{v_1, \ldots, v_{|V|-|\partial G|}\}$ are all the interior points of G, $N(v_i)$ is the set of neighbors of v_i, and $|N(v_i)|$ is the number of neighbors of v_i. A graph is called 3-connected if it remains connected after the removal of any 2 vertices and their incident edges. This is true for most triangular meshes in general. Later, a simple proof of the theorem was suggested by de Verdière et al. [4]. By the theorem, it guarantees a one-to-one and onto map for a simply connected open surface onto a convex planer domain by solving the above systems of linear equations.

3.7.2. Discrete Conformal Embedding. As an alternative to Tutte embedding, one may also compute the discrete conformal embedding of the triangular mesh onto some given domain. The embedding is computed by solving a linear system with "cotangent" weights, first introduced by Pinkall et al. [22] as a linear finite element approximation of the Laplace-Beltrami operator. In the embedding, we require each vertex v_i from the mesh to be mapped to $p_i = (x_i, y_i)$, which satisfies

$$p_i = \sum_{v_j \in N(v_i)} w_{ij} p_j, \tag{3.14}$$

where $N(v_i)$ is the set of neighbors of v_i, and $w_{ij} = \cot \alpha_{ij} + \cot \beta_{ij}$, with α and β given by the angles shown in Figure 3.4.

The sum on the right hand side of (3.14) is often used to define the discrete Laplacian operator on triangular meshes, and can be derived using finite element methods [12]. As our results show, computing the initial map using this method can often speed up our algorithms for uniformization by reducing the number of iterations needed for discrete Beltrami flow, as the initial map is already a conformal map in a discrete sense. However, there is no guarantee that the map is always non-overlapping, which could pose problems to our algorithms if the initial triangular mesh is very irregular.

4. Numerical Algorithms.

In this section, we describe the numerical algorithms we use for surface uniformization, which are based on the various theories on surfaces, their classifications and the distortions of the maps between them. With the results from these theories, our algorithms are able to work on general surfaces with arbitrary topology.

4.1. A Brief Introduction of the Discrete Beltrami Flow Algorithm.

In this subsection, we briefly describe the discrete Beltrami flow algorithm we use for uniformization. First we consider a map from and to the complex plane $\mathbb{C}$. After that, we briefly discuss how the method is generalized to arbitrary surfaces.

Consider a domain $\Omega \in \mathbb{C}^2$. The aim of the Beltrami flow algorithm is to find a quasiconformal map f from Ω to some domain Σ which optimizes some energy $E(\mu)$, where $\mu\colon \mathbb{C} \to \mathbb{C}$ is the Beltrami coefficient of the map. The domain Σ can be fixed or unknown itself depending on the application. In both cases, the algorithm seeks a flow $V\colon \mathbb{C} \to \mathbb{C}$ such that by flowing f using V, i.e., adjust the map f to $g^t(z) = f(z) + tV(z)$ with some appropriate time step t, the energy is decreased.

Suppose the current Beltrami coefficient is μ, and there is a direction $\nu\colon \Omega \to \mathbb{C}$ such that $E(\mu+\nu) < E(\mu)$, we want to compute V such that the Beltrami coefficient of $g^t(z) = f + tV$ is approximately $\mu + t\nu$ for small t. It is shown [25] that V, also called the Beltrami flow, satisfies

$$\frac{\partial V}{\partial \overline{z}} = \left(\frac{\nu}{1-|\mu|^2} \frac{1}{\theta} \right) \circ (f)^{-1}, \tag{4.1}$$

where $\theta = \frac{\overline{p}}{p}$ and $p = \frac{\partial}{\partial z} f(z)$. We compute the best V satisfying this equation using a least squares approach to obtain a flow for adjusting f. For details of the implementation, we refer interested readers to [25], which developed the algorithm for constructing quasiconformal maps between two arbitrary surfaces. For an energy functional E of μ, we can set ν to be the gradient decent direction for example, and compute the flow V giving the required adjustment in μ. On discrete surfaces, the least square energy can be written as a sum of squares of discrete derivatives computed on every triangle. This method is called the discrete Beltrami flow algorithm, and was used to produce results for simply connected closed surfaces in that paper. The square of the L^2-norm of the Beltrami differential was used as the energy functional and produced excellent results. Other energy functionals of μ can be used as well, which is one of the advantages of our algorithms.

4.2. Initialization for Genus One Surfaces.

To construct an initial map from a genus one surface M onto the fundamental domain $[0,1] \times [0,1] \in \mathbb{R}^2$ with initial generators $(1,0), (0,1) \in \mathbb{R}^2$, we first select a point p from the vertices of the triangular mesh of M, to be mapped to the 4 corners of the fundamental domain. We pick 2 homotopically different loops with p as endpoints on the mesh to represent the homological basis of the first homology group of the mesh, which will be mapped onto the 2 pairs of opposite sides respectively. These loops can be picked by hand or can be constructed by some graph searching algorithms, e.g. [5]. After cutting up the mesh along these loops, we obtain a graph representation of the cut open mesh, which is open and simply connected. Using Tutte embedding or discrete conformal

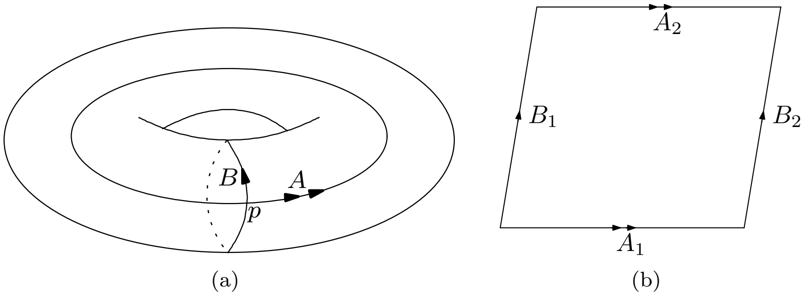


FIG. 4.1. *The cuts on a torus and how the cut open torus is mapped onto a fundamental domain. (a) shows a torus with* 2 *cuts made along* 2 *homotopically different loops at their intersection. (b) shows how the cut open torus is then mapped onto a fundamental domain. The corresponding sides on the torus and the fundamental domain are marked with the same number of arrows.*

embedding, we can easily map M onto $[0,1]\times[0,1]$ with 2 pairs of sides identified for consistency. This completes our initialization for genus one surfaces.

4.3. Discrete Beltrami Flow for Genus One Surfaces. Suppose we have a parametrization from a genus one surface M onto a fundamental domain $\Omega \in \mathbb{R}^2$, with generators $(1,0)$ and (a,b) in $\mathbb{R}^2$ as discussed in 4.2. Suppose $\partial\Omega = A_1 \cup B_2 \cup A_2 \cup B_1$, where $A_2 = A_1 + a + b\sqrt{-1}$ and $B_2 = B_1 + 1$. We want to flow this domain in $\mathbb{R}^2$ so that the generator $(a(t), b(t))$ is changing as a function of time and A_1, A_2 move consistently with respect to this change of generator, i.e.,

$$A_2 = A_1 + a(t) + b(t)\sqrt{-1}. \tag{4.2}$$

Figure 4.1 illustrates the cuts on the torus and how the cut open torus is mapped onto a fundamental domain with the correspondence described.

For each point p_1 on A_1 and its corresponding point p_2 on A_2, by (4.2), we have the relation

$$p_2(t) = p_1(t) + a(t) + b(t)\sqrt{-1}. \tag{4.3}$$

Therefore, on A_1 and A_2, we have the following condition for the flow V:

$$V(p_2) = V(p_1) + \frac{d}{dt}a(t) + \frac{d}{dt}b(t)\sqrt{-1}. \tag{4.4}$$

On B_1 and B_2, since the generator $(1,0)$ is not changing, we have

$$q_2(t) = q_1(t) + 1 \tag{4.5}$$

for all q_1 on B_1 and its corresponding point q_2 on B_2. Therefore, on B_1 and B_2, we have

$$V(q_2) = V(q_1). \tag{4.6}$$

To adjust the diffeomorphism according to a given Beltrami coefficient ν, we minimize the following energy functional

$$\int_\Omega \left| \frac{\partial}{\partial \bar{z}} V(z) - \nu(z) \right|^2 dx\,dy \tag{4.7}$$

with the above constraints (4.4) and (4.6) along the boundaries strictly imposed as boundary conditions for V.

We summarize the algorithm for genus one surfaces in Algorithm 1 and leave the discussion about the optimal time step to Subsection 4.6.

Algorithm 1 Compute the uniformization for a genus one surface

```
Require: A genus one surface represented by a triangular mesh $M = \{P, E, F\}$; Pick
    a vertex $p \in P$ to be mapped to the 4 corners of the fundamental domain; Pick 2
    loops $L_1$, $L_2$ through $p$ representing the homological basis of $M$ formed by points
    in $P$ and edges in $E$;
  Using Tutte embedding or discrete conformal embedding, compute a map $\phi$ from
    the open surface cut by $L_1$ and $L_2$ onto $[0,1] \times [0,1]$ with $\phi^{-1}([0,1] \times 0) = L_1$,
    $\phi^{-1}([0,1] \times 1) = L_1$, $\phi^{-1}(0 \times [0,1]) = L_2$ and $\phi^{-1}(1 \times [0,1]) = L_2$;
  Set tolerance TOL;
  Set generators $(a, b) = (0, 1)$;
  repeat
    Compute the the least squares flow $V$ in (4.7) with constraints (4.4) and (4.6);
    Set $\phi_{\text{new}}(z) := \phi(z) + tV(\phi(z))$, where $t$ is an optimally chosen time step;
    Set $(a, b) = \phi_{\text{new}}(0, 1)$;
    Set $\phi$ to $\phi_{\text{new}}$;
  until $\max_{z \in P} t \cdot |V(z)| < \text{TOL}$.
```

4.4. Initialization for High Genus Surfaces. For a high genus surface M with genus greater than 1, we can also construct an initial map from M onto the fundamental polygon in D. However, since the geodesics in D are not straight lines in Euclidean sense, the fundamental polygon is not convex in Euclidean sense under the Poincaré model. To solve this problem, we note that the geodesics are straight lines in Euclidean sense under the Beltrami-Klein model of the hyperbolic disk, and the coordinates in these models are related by

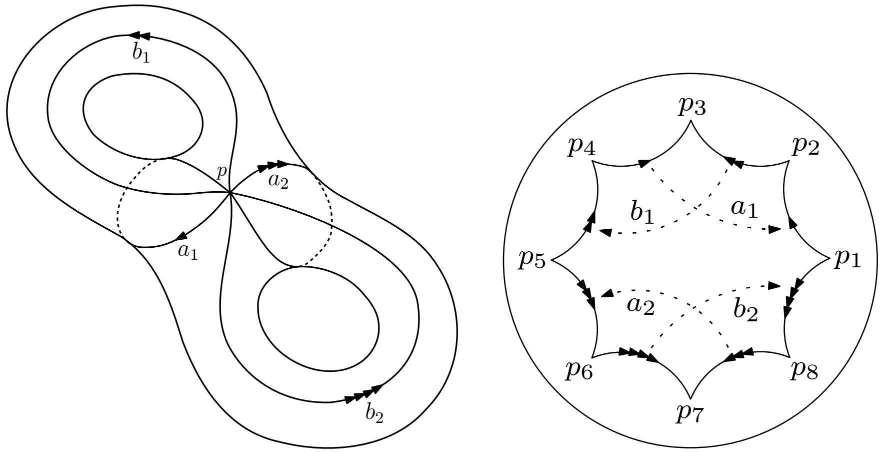
$$s = \frac{2u}{1 + u \cdot u}. \tag{4.8}$$

Therefore we can first embed M onto the fundamental polygon under the Beltrami-Klein model, which is convex in Euclidean sense, then map the resulting initialization back to the Poincaré model.

Similar to the initialization of genus one surfaces, we first pick a point p from the vertices of the triangular mesh of M, to be mapped to the corners of the fundamental polygon. Then we select $2g$ homotopically different loops with p as endpoints on the mesh that represent the homological basis of the first homology group of the mesh. Once the loops are chosen, the mesh can be cut along these loops to obtain a simply connected open surface. Then Tutte embedding or discrete conformal embedding can be used to map M onto the fundamental polygon under the Beltrami-Klein model. Finally, we transform the coordinates of the image of M to get its image under the Poincaré model by the formula

$$u = \frac{s}{1 + \sqrt{1 - s \cdot s}}. \tag{4.9}$$

This completes the initialization for high genus surfaces. Note that the error introduced in the conversion between the two models does not matter because the initial-



(a) (b)

FIG. 4.2. *The cuts on a genus 2 surface and how the cut open surface is mapped onto a fundamental polygon. (a) shows a genus 2 surface with 4 cuts made along 4 homotopically different loops at their intersection. (b) shows how the cut open surface is then mapped onto a fundamental polygon. With such polygon, we immediate obtain the generators a_1, a_2, b_1 and b_2 of the Fuchsian group of the surface. The corresponding sides on the surface and the fundamental polygon are marked with the same number of arrows.*

ization is not a conformal map. The distortion will be corrected in the Beltrami flow algorithm.

As an example, in Figure 4.2, we illustrate the cuts of a genus 2 surface and how the cut open surface is mapped onto a fundamental polygon using the initialization described.

4.5. Discrete Beltrami Flow for High Genus Surfaces. Consider points $p_1(t)$, $p_2(t)$ on ∂D, which correspond to each other by a Möbius transformation ψ^t, which is also changing with the flow. To move p_1 and p_2 consistently in each iteration, we derive the linear constraints that the velocities of the points and the change of parameters of the Möbius transformation have to satisfy. Suppose $p_1(t) = (x_1(t), y_1(t))$ and $p_2(t) = (x_2(t), y_2(t))$, and let ψ^t be defined by

$$\psi^t(z) = e^{i\phi(t)} \frac{z - a(t)}{1 - \overline{a(t)} z}, \tag{4.10}$$

we have the relation

$$\psi^t(p_1(t)) = p_2(t), \tag{4.11}$$

or

$$e^{i\phi(t)} \frac{p_1(t) - a(t)}{1 - \overline{a(t)} p_1(t)} = p_2(t), \tag{4.12}$$

By differentiating the above equation, one gets the linear relation of changes among $p_1(t)$, $p_2(t)$ and the parameters $\phi(t)$ and $a(t)$,

$$\dot{p_2} = \dot{\psi}^t(p_1) = p_2 i\dot{\phi} + \frac{e^{i\phi} + p_2\overline{a}}{1 - \overline{a}p_1}\dot{p_1} - \frac{e^{i\phi}}{1 - \overline{a}p_1}\dot{a} + \frac{p_1 p_2}{1 - \overline{a}p_1}\dot{\overline{a}}. \tag{4.13}$$

If p_1 and p_2 are related perfectly by (4.12), then we can compute the least square Beltrami flow with the above linear constraint applied to every pairs of corresponding points. However, the corresponding points and the parameters of the corresponding Möbius transformation may not satisfy the relation (4.12) initially or during the evolution, since the adjustment of the corresponding points is made in the Euclidean sense after the vector field is computed. Therefore we use a relaxation way to enforce the relation during the evolution by

$$\dot{p_2} - \dot{\psi}(p_1) = -\alpha(p_2 - \psi(p_1)), \tag{4.14}$$

where α is some relaxation parameter. In our experiment we use $\alpha = 1$, because ideally, a time step of $t = 1$ should adjust the diffeomorphism to a uniformizing map. This is also equivalent to adding a Lagrange multiplier to the energy functional of the diffeomorphism to enforce the relation (4.12) in a least square sense. So we use the following constraint for the velocity field at the vertices of the target fundamental polygons in D:

$$\dot{p_2} + p_2 - \psi(p_1) = p_2 i\dot{\phi} + \frac{e^{i\phi} + p_2\overline{a}}{1 - \overline{a}p_1}\dot{p_1} - \frac{e^{i\phi}}{1 - \overline{a}p_1}\dot{a} + \frac{p_1 p_2}{1 - \overline{a}p_1}\dot{\overline{a}}. \tag{4.15}$$

Another important constraint in our algorithm is to keep the generator relations valid. In other words, the fundamental domain has to morph in a way such that they can tessellate the whole hyperbolic disk. This requires all pairs of corresponding edges to have equal length, and the angle sum of the fundamental polygon to be 2π. Interestingly, these constraints are automatically enforced by (4.12). This is because (4.12) guarantees that each pair of corresponding sides is related by a Möbius transformation, which is an isometry. Therefore the copies of the corners of the fundamental domain meet at a point (e.g. one of its corners) with all pairs of its sides consistently matched in length. Since the angle around every point in the hyperbolic disk is 2π, the sum of the angles of the corners has to be 2π as well. This guarantees that the generator relation (3.2) is satisfied, and no extra constraint other than (4.12) is needed.

We summarize the algorithm for high genus surfaces in Algorithm 2 and leave the discussion about the optimal time step to the following subsection.

4.6. Choosing the Optimal Time Step.

In both of the algorithms above, the time step should be chosen to lead to a fast decay of the least square energy of the Beltrami differential $\mu(z)\frac{\overline{dz}}{dz}$ of surface M:

$$E\left(\mu(z)\frac{\overline{dz}}{dz}\right) = \int_M |\mu(z)|^2 \, dA. \tag{4.16}$$

Note that since the modulus of $\mu(z)$ does not depend on the coordinates chosen to represent the Beltrami differential, the above definition makes sense.

After computing the vector field V using Beltrami flow that adjusts the Beltrami differential of the surface map in the direction of the steepest descent of (4.16), the diffeomorphism with an adjustment of size t is given by $\phi_{\text{new}}(z) = \phi(z) + tV(\phi(z))$. Therefore the Beltrami differential also depends on t. We can simplify the notation and write

$$E(t) = E\left(\mu(z,t)\frac{\overline{dz}}{dz}\right) = \int_M |\mu(z,t)|^2 \, dA, \tag{4.17}$$

Algorithm 2 Compute the uniformization for a high genus surface

```
Require: A genus $g$ surface represented by a triangular mesh $M = \{P, E, F\}$, with $g \geq 2$; Pick a vertex $p \in P$ to be mapped to the $4g$ corners of the fundamental polygon in the hyperbolic disk $D$; Pick $2g$ loops $L_1, \dots, L_{2g}$ through $p$ representing the homological basis of $M$ formed by points in $P$ and edges in $E$;
  Using Tutte embedding or discrete conformal embedding, compute a map $\phi$ from the open surface cut by $L_1, \dots, L_{2g}$ onto the standard $4g$-gon in $D = \text{int}(D) \cup A_1 \cup B_1 \cup A_1^{-1} \cup B_1^{-1} \cup \cdots \cup A_g \cup B_g \cup A_g^{-1} \cup B_g^{-1}$, with $\phi^{-1}(A_i) = L_{2i-1}$, $\phi^{-1}(A_i^{-1}) = L_{2i-1}$, $\phi^{-1}(B_i) = L_{2i}$ and $\phi^{-1}(B_i^{-1}) = L_{2i}$ for $i = 1, \dots, g$;
  Set tolerance TOL;
  Set generators $g_i(z) = e^{\sqrt{-1}\phi_i} \frac{z-a_i}{1-\overline{a_i}z}$;
  repeat
    Compute the the least squares flow $V$ in (4.7) with the constraints defined by (4.15);
    Set $\phi_{\text{new}}(z) := \phi(z) + tV(\phi(z))$, where $t$ is an optimally chosen time step;
    Set $\phi$ to $\phi_{\text{new}}$;
  until $\max_{z \in P} t \cdot |V(z)| < \text{TOL}$.
```

where we have abused the notation and assumed we have a global parametrization in z. This is not essential as our computation is done locally. The first and second derivatives of $|\mu(z,t)|^2$ can be computed on each triangle of the triangulated M. Each such triangle can be considered as a plane with given local coordinates (x, y), and each triangle of the target domain can also be given local coordinates (u, v). Therefore the definition of the Beltrami coefficient with local coordinates (3.8) can be used. The flow V can also be represented locally as $V(x, y) = V_1(x, y) + \sqrt{-1} \cdot V_2(x, y) \in \mathbb{C}$. Let $f_0(x, y) = u(x, y) + \sqrt{-1} \cdot v(x, y)$ be the local representation of the current surface map. Then $f(x, y, t) = u(x, y) + tV_1(x, y) + \sqrt{-1} \cdot [v(x, y) + tV_2(x, y)]$ gives the local representation of the surface map at time t after applying the flow. Therefore locally, $\mu(z,t)$ is just a complex fraction in terms of u, v, V_1 and V_2. Hence, $\frac{d}{dt}|\mu(z)|^2$ and $\frac{d^2}{dt^2}|\mu(z)|^2$ can be computed easily. We have put the derivation of these two formulas in detail in the Appendix.

With $\frac{d}{dt}|\mu(z)|^2$ and $\frac{d^2}{dt^2}|\mu(z)|^2$ computed locally on each face, the first derivative of $E(t)$ is given by

$$E'(t) = \int_M \frac{d}{dt}|\mu(z,t)|^2 \, dA. \tag{4.18}$$

Also, the second derivative of $E(t)$ is given by

$$E''(t) = \int_M \frac{d^2}{dt^2}|\mu(z,t)|^2 \, dA. \tag{4.19}$$

The integrals for both $E'(t)$ and $E''(t)$ can be written as summations over all faces of M.

For small t, $E(t)$ can be approximated by

$$E(t) = E(0) + E'(0)t + \frac{1}{2}E''(0)t^2. \tag{4.20}$$

Using the Newton's method, the optimal time step t_{optimal} is the time step t such that the first derivative of the above approximation becomes 0, which means the energy attains a local minimum. The optimal time step is then given by

$$t_{\text{optimal}} = -\frac{E'(0)}{E''(0)}. \tag{4.21}$$

Another constraint of the time step is that the Jacobian J of the map must be positive on M. Locally, the Jacobian of the resulting map with time step t in direction V can be computed on each face as a quadratic polynomial in t. Similar to the above discussion, on a face of M, we can write the current map as $f_0(x,y) = u(x,y) + \sqrt{-1} \cdot v(x,y)$ and the flow as $V(x,y) = V_1(x,y) + \sqrt{-1} \cdot V_2(x,y) \in \mathbb{C}$ using local coordinates. The local representation of the surface map at time t is given by $f(x,y,t) = u(x,y) + tV_1(x,y) + \sqrt{-1} \cdot [v(x,y) + tV_2(x,y)]$. The Jacobian of the face at time t is given by

$$\begin{aligned} J(x,y,t) &= \left(\frac{\partial u}{\partial x} + t\frac{\partial V_1}{\partial x}\right)\left(\frac{\partial v}{\partial y} + t\frac{\partial V_2}{\partial y}\right) - \left(\frac{\partial u}{\partial y} + t\frac{\partial V_1}{\partial y}\right)\left(\frac{\partial v}{\partial x} + t\frac{\partial V_2}{\partial x}\right) \\ &= \left(\frac{\partial u}{\partial x}\frac{\partial v}{\partial y} - \frac{\partial u}{\partial y}\frac{\partial v}{\partial x}\right) + t\left(\frac{\partial u}{\partial x}\frac{\partial V_2}{\partial y} + \frac{\partial v}{\partial y}\frac{\partial V_1}{\partial x} - \frac{\partial u}{\partial y}\frac{\partial V_2}{\partial x} - \frac{\partial v}{\partial x}\frac{\partial V_1}{\partial y}\right) \\ &\quad + t^2\left(\frac{\partial V_1}{\partial x}\frac{\partial V_2}{\partial y} - \frac{\partial V_1}{\partial y}\frac{\partial V_2}{\partial x}\right), \end{aligned} \tag{4.22}$$

which is a quadratic polynomial in t.

Therefore, on each triangle T, we can set the maximal time step $t_{\max,T}$ we can take to be the smallest positive root of the above polynomial in t for that triangle. This is the smallest time step that causes the triangle to degenerate and then overlap for any bigger time step. Therefore we define the global maximum time step allowed as

$$t_{\max} = \min_{T \in F} t_{\max,T}. \tag{4.23}$$

In our algorithms, we choose our time step t_{chosen} as

$$t_{\text{chosen}} = \min(t_{\text{optimal}}, t_{\max}/2). \tag{4.24}$$

This ensures fast convergence of our algorithms and guarantees that the resulting map is not close to overlapping.

5. Results and Analysis.

In this section, we present numerical results of our algorithms. Before Section 5.5, Tutte embedding is used for the construction of initial map. All computations are performed using MATLAB on a mobile machine equipped with a dual-core 2.9 GHz processor and 8 GB of RAM. With a more efficient implementation in C/C++, the computation time can be further reduced.

5.1. Uniformization for Genus Zero Surfaces.

We compute the uniformization of 6 genus zero surfaces onto the unit sphere. The complexity of the triangulated surfaces ranges from 6002 vertices to 128930 vertices. As shown in Table 5.1, except for the 2 largest meshes with more than 100K vertices, the uniformization for other genus zero surfaces of up to 30K can be computed in a few seconds.

5.2. Uniformization for Genus One Surfaces.

We test our algorithm on the computation for genus one surfaces, which is the simplest case for surfaces with genus $g \geq 1$. The target domain is some parallelogram in the plane, and only 1 generator

Table 5.1

The results of our algorithm for the uniformization of genus zero surfaces, with Tutte embedding initialization.

Surface	**# Vertices**	**# Triangles**	**# Iterations**	**Time Taken (s)**
Human Brain	6002	12000	7	2.6923
Fandisk	6475	12946	6	2.6897
Octa Flower	7919	15834	20	9.9278
Fish	29498	58992	14	16.7971
Fish 120K	120069	240134	22	222.0254
Armadillo	128930	257856	34	336.6296

Table 5.2

The results of our algorithm for the uniformization of genus one surfaces, with Tutte embedding initialization.

Surface	**# Vertices**	**# Triangles**	**# Iterations**	**Time Taken (s)**
Rocker Arm	9397	18794	13	4.0616
Kitten	24956	49912	8	7.9141
Bumpy Torus	16815	33630	12	7.3783

with 2 real parameters are needed. The results for 3 examples are shown in Table 5.2. The time required by our algorithm for genus one cases is comparable to similar sized meshes for genus zeros cases. The final uniformization of these surfaces and their uniformization domains are shown in Figure 5.1.

5.3. Uniformization for Genus Two Surfaces. For genus two surfaces, since the target uniformization domain is the hyperbolic disk, and the relations between the sides of the fundamental polygon are non-linear, we restrict out maximum possible step size to $0.2 \cdot t_{\max}$ to prevent points from going outside the disk. Our algorithm works well for both examples. When the algorithm converges, the relative error of the equations describing the generator relations (4.12) approaches machine precision. This indicates that our algorithm successfully works on the hyperbolic disk. The results are shown in Table 5.3. The final uniformization of these surfaces and their uniformization domains are shown in Figure 5.2.

5.4. Comparison with Other Methods. We compare our algorithms with 2 other existing methods, namely the harmonic map method and methods based on Ricci flow. We compare the results of these methods in terms of the running time and the angular distortion of the final parametrization.

For the harmonic map method, we compute the harmonic map from 2 genus zero surface onto a sphere. Theoretically, such map always exists and is unique and conformal if we fix 3 points on the surface. The harmonic maps are computed using gradient descent for the Octa Flower mesh and the Fish mesh, both genus 0 surfaces. With the same initial maps as used in our algorithm and the same stopping criteria for both meshes, the running time for Octa Flower was 3.7649 seconds and that for Fish was 40.2178 seconds. This shows that as the mesh size grows larger, our algorithm outperforms the harmonic map method.

We also tested the Ricci flow algorithms for genus one and high genus surfaces from the RiemannMapper programs publicly available[1]. The programs were written

[1] http://www.cs.sunysb.edu/ gu/software/RiemannMapper/index.html

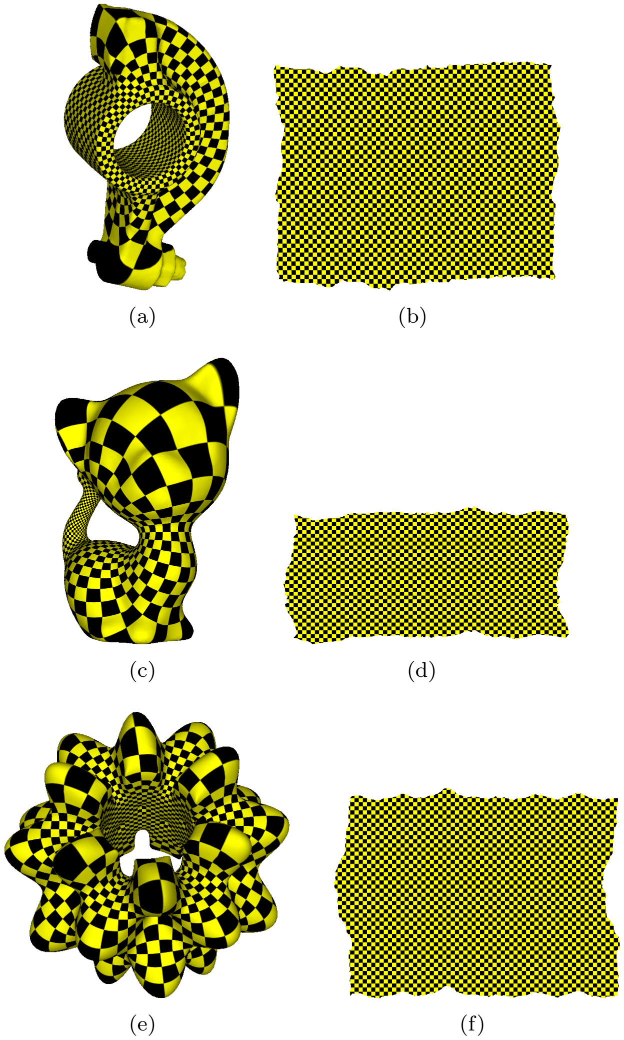


Fig. 5.1. *The uniformization of* 3 *genus one surfaces and their fundamental domains. (a)(b) Rocker Arm. (c)(d) Kitten. (e)(f) Bumpy Torus.*

Table 5.3

The results of our algorithm for the uniformization of genus two surfaces, with Tutte embedding initialization.

Surface	**# Vertices**	**# Triangles**	**# Iterations**	**Time Taken (s)**
Amphora	9014	10832	40	31.3576
Figure Eight	12286	24576	32	22.7450

in C++ and call MATLAB at runtime for matrix computations. In this test, 2 genus one surfaces and 1 genus two surfaces were used, and each of them has around 10K vertices. In particular, the genus two surface of Figure Eight used was the same as our test examples. We found that the uniformization of these 3 surfaces can each be computed in 5 to 6 seconds. This suggests that the Ricci flow algorithms are highly efficient due to the use of Newton's method for energy minimization. From the results

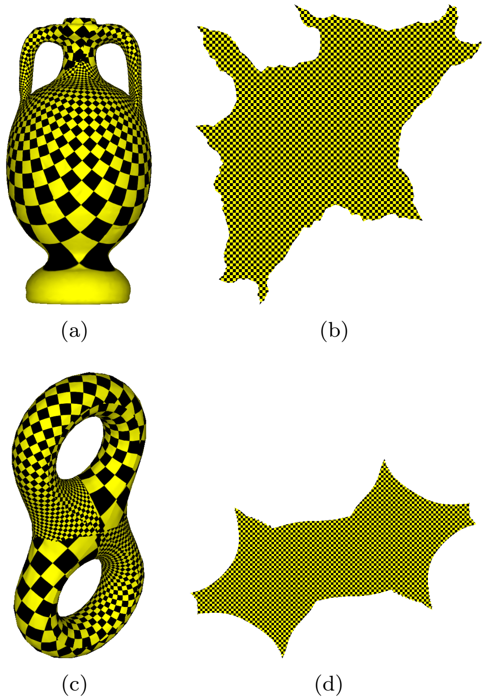


Fig. 5.2. *The uniformization of* 2 *genus two surfaces and their fundamental domains. (a)(b) Amphora. (c)(d) Figure Eight.*

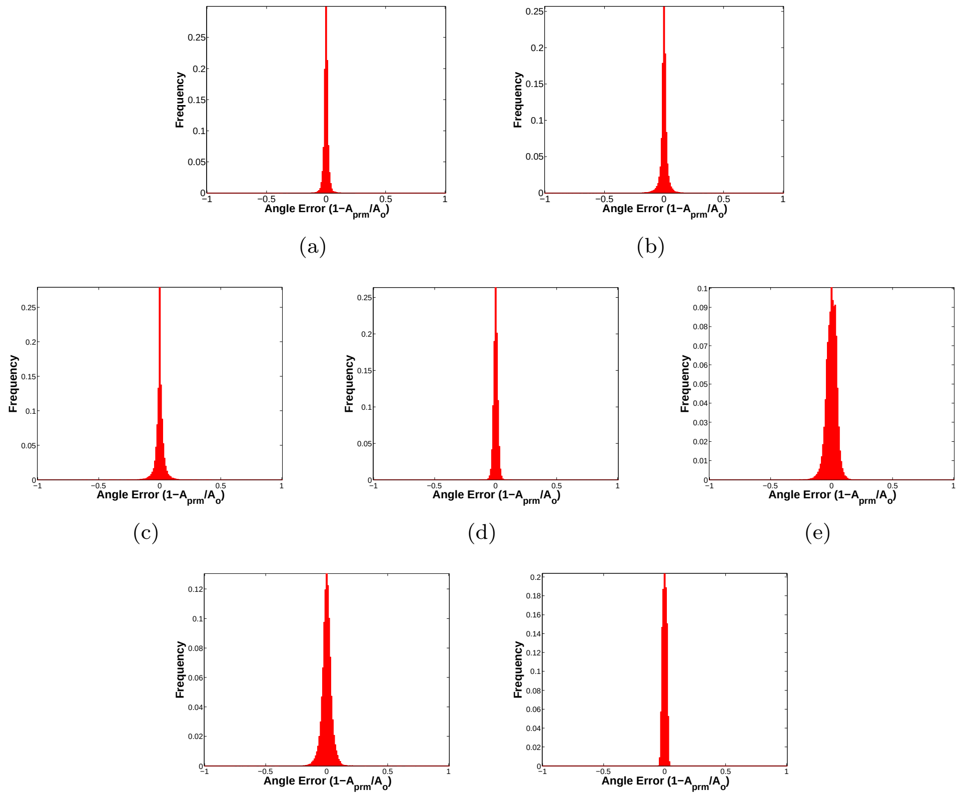
of our algorithms, it can be seen that our algorithm works reasonably fast, with the Kitten, a mesh with 25K vertices, taking only 8 seconds. For high genus surfaces, our algorithm runs slower and takes nearly 23 seconds on the figure eight mesh of 12K vertices. There are two factors here. One is because our code is in MATLAB. The other is because we used the step size $0.2 \cdot t_{\max}$ to prevent points from going outside the hyperbolic disk. We hope to improve its efficiency by using a more flexible step size in the future.

To compare the distribution of angular distortion from different methods, we compute the distribution of distortions in each triangle of the surface mesh and the parametrized mesh. Denote the angle from the face of the original surface by A_{o} and that from the face of the parametrized mesh by A_{prm}, we compute the distribution of the quantity

$$1 - \frac{A_{\text{prm}}}{A_{\text{o}}}. \tag{5.1}$$

If the value of the above quantity is close to 0, it means the parametrized angle is close to the original angle, which means a small angular distortion. By comparing the angular distortion using different algorithms, we can conclude which method achieves the best result in preserving conformality.

We plot the angular distortions of the results from our algorithms in Figure 5.3, and results from other algorithms in Figure 5.4. It can be seen that our method produces very little angular distortion for surfaces of all topologies consistently. On the other hand, the widely dispersed histograms of the results from harmonic maps



(a) (b)

(c) (d) (e)

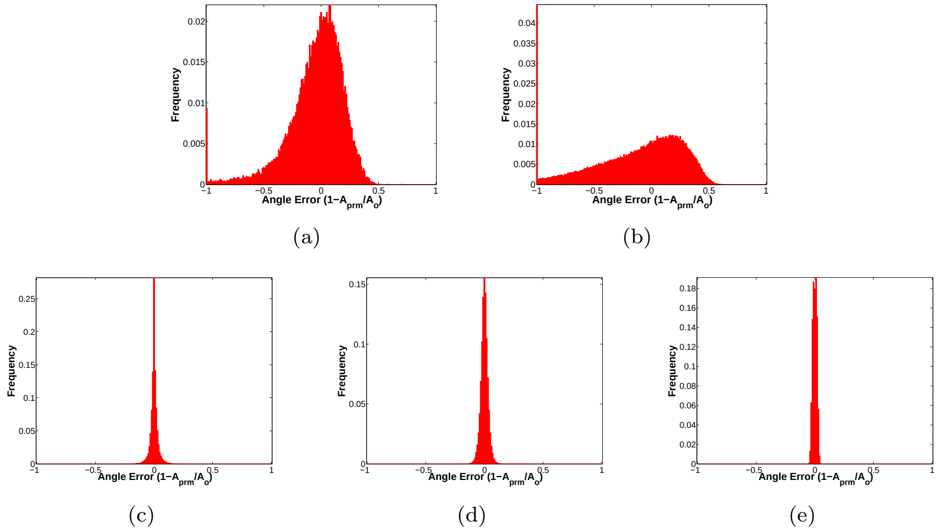
(f) (g)

Fig. 5.3. *The distribution of angular error of the results by our algorithms. (a) Octa Flower. (b) Fish. (c) Rocker Arm. (d) Kitten. (e) Bumpy Torus. (f) Amphora. (g) Figure Eight.*

show serious angular distortions. Our algorithm produces similar results to those from Ricci flow algorithm which is considered the current state of the art. Also, the serious angular errors show that the harmonic map may not be a good characterization for conformal map for discrete triangulated surfaces.

5.5. Improvement from Discrete Conformal Initialization. We also tested our algorithms on all meshes using the discrete conformal initialization discussed in Section 3.7.2. While there is no guarantee that the initial map is always non-overlapping, most meshes can be properly initialized onto the initial domain of parametrization. We also find that speed of our algorithms can be significantly improved when discrete conformal embedding can be used for initial map. The most impressive result comes from the uniformization of the fish mesh with 120K vertices. We are able to reduce the 22 iterations it takes using Tutte embedding to only 5 iterations using discrete conformal initialization. This suggests that we should initialize using the discrete conformal map whenever possible. The results for genus zero, one and two surfaces are shown in Tables 5.4, 5.5 and 5.6 respectively, where N/A means discrete conformal initial map does not work with Beltrami flow.

5.6. Performance on Meshes with Sharp Edges. Finally, we test our algorithms on meshes with very sharp triangles. The meshes used in our experiment



(a)

(b)

(c)

(d)

(e)

Fig. 5.4. *The distribution of angular error of the results by other algorithms. (a) Octa Flower by harmonic map. (b) Fish by harmonic map. (c) Rocker Arm by Ricci flow. (d) Kitten by Ricci flow. (e) Figure Eight by Ricci flow.*

Table 5.4

The results of our algorithm for the uniformization of genus zero surfaces, with discrete conformal initialization.

Surface	**# Vertices**	**# Triangles**	**# Iterations**	**Time Taken (s)**
Human Brain	6002	12000	5	1.9936
Fandisk	6475	12946	6	2.6847
Octa Flower	7919	15834	6	3.2029
Fish	29498	58992	N/A	N/A
Fish 120K	120069	240134	5	68.4591
Armadillo	128930	257856	8	129.0380

are shown in Figure 5.5. Meshes of this form is very common in computer-aided design (CAD) applications, where very sharp triangles occur because some vertex is connected with many neighbors, as apparent in the figures. The main obstacle of the uniformization of these surfaces comes from the difficulty of finding a good initial map. The initial maps may contain triangles with very different sizes, which causes the algorithms to be less stable. However, once a good initial map is found, our algorithms work as expected. The results of our algorithms on the example meshes are shown in Table 5.7.

6. Conclusion. In this paper, we introduced a novel algorithm based on discrete Beltrami flow for the computation of surface uniformizations. By computing a vector field with proper constraints enforced for the adjustment of different generators, we successfully computed the uniformization for surfaces with genus $g \geq 1$. Computation results show that our method a can successfully construct the uniformization map and achieve both good accuracy and efficiency suggesting that parametrization by minimizing the L^2-norm of the Beltrami differential could be used as an alternative good characterization for discrete surface uniformization.

Table 5.5

The results of our algorithm for the uniformization of genus one surfaces, with discrete conformal initialization.

Surface	**# Vertices**	**# Triangles**	**# Iterations**	**Time Taken (s)**
Rocker Arm	9397	18794	7	5.9869
Kitten	24956	49912	6	6.0294
Bumpy Torus	16815	33630	4	2.4281

Table 5.6

The results of our algorithm for the uniformization of genus two surfaces, with discrete conformal initialization.

Surface	**# Vertices**	**# Triangles**	**# Iterations**	**Time Taken (s)**
Amphora	9014	10832	N/A	N/A
Figure Eight	12286	24576	27	18.8043

Appendix. In this part, we derive the derivatives of the Beltrami coefficient with respect to the time, when the local representation of the surface map $f_0(x,y) = u(x,y) + \sqrt{-1} \cdot v(x,y)$ is adjusted by the flow $V(x,y) = V_1(x,y) + \sqrt{-1} \cdot V_2(x,y)$, also given by a local representation. We have used (x,y) and (u,v) to represent the coordinates of the source and target surfaces respectively. Therefore $f(x,y,t) = u(x,y) + tV_1(x,y) + \sqrt{-1} \cdot [v(x,y) + tV_2(x,y)]$ gives the local representation of the surface map at time t after applying the flow. Our target is to compute $\frac{d}{dt}|\mu(z)|^2$ and $\frac{d^2}{dt^2}|\mu(z)|^2$ from the given information.

Write $f(z,t) = f(x,y,t) = f_0(z) + tV(z)$, where $z = x + \sqrt{-1} \cdot y$ and we have treated f and V as complex-valued functions on the complex plane. Then the Beltrami coefficient $\mu(z,t)$ at $z \in \mathbb{C}$ and time t is given by

$$\mu(z,t) = \frac{f_{0\overline{z}} + tV_{\overline{z}}}{f_{0z} + tV_z}. \tag{6.1}$$

Then

$$\begin{aligned} \mu'(z,t) &= \frac{(f_{0z} + tV_z)V_{\overline{z}} - (f_{0\overline{z}} + tV_{\overline{z}})V_z}{(f_{0z} + tV_z)^2} \\ &= \frac{V_{\overline{z}} - \mu(z,t)V_z}{f_{0z} + tV_z}. \end{aligned} \tag{6.2}$$

Also

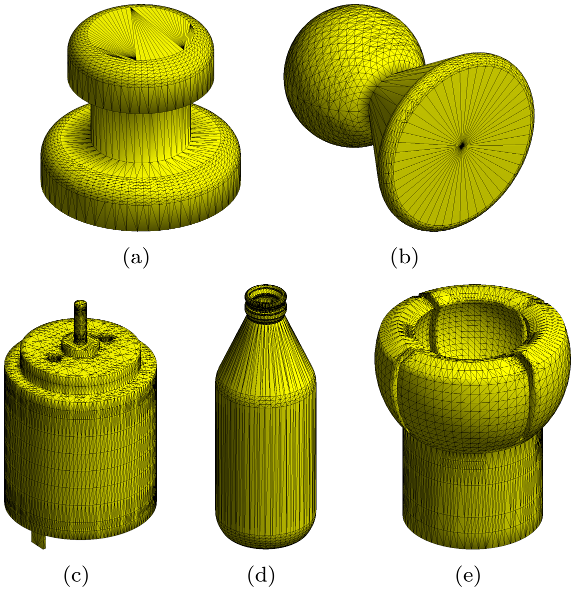
$$\mu''(z,t) = \frac{(f_{0z} + tV_z)(-\mu'(z,t)V_z) - (V_{\overline{z}} - \mu(z,t)V_z)V_z}{(f_{0z} + tV_z)^2}. \tag{6.3}$$

Then we can easily compute the derivatives of $|\mu(z,t)|^2$ with respect to t. For the first derivative,

$$\begin{aligned} \frac{d}{dt}|\mu(z,t)|^2 &= \frac{d}{dt}\left(\mu(z,t)\overline{\mu(z,t)}\right) \\ &= \mu'(z,t)\overline{\mu(z,t)} + \mu(z,t)\overline{\mu'(z,t)}. \end{aligned} \tag{6.4}$$

For the second derivative,

$$\frac{d^2}{dt^2}|\mu(z,t)|^2 = \mu''(z,t)\overline{\mu(z,t)} + 2\mu'(z,t)\overline{\mu'(z,t)} + \mu(z,t)\overline{\mu''(z,t)}. \tag{6.5}$$


(a) (b)

(c) (d) (e)

Fig. 5.5. *The examples used to test our algorithms on meshes with sharp triangles: (a) Magnet Gromet, (b) Pawn, (c) Motor, (d) Bottiglia, and (e) Ball Joint.*

Table 5.7
The results of our algorithm for the uniformization of surfaces with sharp triangles, with Tutte embedding initialization.

Surface	**# Vertices**	**# Triangles**	**# Iterations**	**Time Taken (s)**
Magnet Grommet	1924	3844	23	1.5320
Pawn	2289	4574	22	1.8831
Motor	4430	8856	17	2.5983
Bottiglia	4827	9650	23	4.1106
Ball Joint	6689	13374	21	5.0306

This completes the derivation of the formulas.

REFERENCES

[1] William Abikoff, *The uniformization theorem*, The American Mathematical Monthly, 88 (1981), pp. 574+.

[2] Lars V. Ahlfors, *Lectures on Quasiconformal Mapping*, vol. 38 of University Lecture Series, American Mathematical Society, 2006.

[3] B. Chow and F. Luo, *Combinatorial ricci flows on surfaces*, ArXiv Mathematics e-prints, (2002).

[4] ric Colin de Verdire, Michel Pocchiola, and Gert Vegter, *Tutte's barycenter method applied to isotopies.*, Comput. Geom., 26 (2003), pp. 81–97.

[5] Jeff Erickson, *Shortest non-trivial cycles in directed surface graphs*, in Proceedings of the Twenty-seventh Annual Symposium on Computational Geometry, SoCG '11, New York, NY, USA, 2011, ACM, p. 236243.

[6] Louis Funar, *Lectures on fuchsian groups and their moduli*, 2004.

[7] Frederick P. Gardiner and Nikola Lakic, *Quasiconformal Teichmller Theory*, vol. 76 of Mathematical Surveys and Monographs, American Mathematical Society, 1999.

[8] Xianfeng Gu, Yalin Wang, Tony F. Chan, Paul M. Thompson, and Shing-tung Yau, *Genus zero surface conformal mapping and its application to brain surface mapping*, IEEE

Transactions on Medical Imaging, 23 (2004), p. 949958.

[9] Xianfeng Gu and Shing-Tung Yau, *Surface Classification Using Conformal Structures*, 2003.

[10] Steven Haker, Sigurd Angenent, Allen Tannenbaum, Ron Kikinis, Guillermo Sapiro, and Michael Halle, *Conformal surface parameterization for texture mapping*, IEEE Transactions on Visualization and Computer Graphics, 6 (2000), p. 181189.

[11] M.K. Hurdal and K. Stephenson, *Discrete conformal methods for cortical brain flattening.*, Neuroimage, (2008).

[12] A. Iserles, *A first course in the numerical analysis of differential equations*, Cambridge texts in applied mathematics, Cambridge University Press, Cambridge, New York, 1996. lccopycat.

[13] Miao Jin, Junho Kim, and Xianfeng David Gu, *Discrete surface ricci flow: theory and applications*, in Proceedings of the 12th IMA international conference on Mathematics of surfaces XII, Berlin, Heidelberg, 2007, Springer-Verlag, p. 209232.

[14] Miao Jin, Yalin Wang, Shing-Tung Yau, and Xianfeng Gu, *Optimal global conformal surface parameterization*, in Proceedings of the conference on Visualization '04, VIS '04, Washington, DC, USA, 2004, IEEE Computer Society, p. 267274.

[15] Yaron Lipman, *Bounded distortion mapping spaces for triangular meshes*, ACM Trans. Graph., 31 (2012), p. 108:1108:13.

[16] Yaron Lipman, Vladimir G. Kim, and Thomas A. Funkhouser, *Simple formulas for quasiconformal plane deformations*, ACM Trans. Graph., 31 (2012), p. 124:1124:13.

[17] Yaron Lipman, J. Puente, and Ingrid Daubechies, *Conformal wasserstein distance: II. computational aspects and extensions.*, Math. Comput., 82 (2013).

[18] Lok Ming Lui, Ka Chun Lam, Tsz Wai Wong, and Xianfeng Gu, *Beltrami representation and its applications to texture map and video compression*, CoRR, abs/1210.8025 (2012).

[19] Lok Ming Lui, Tsz Wai Wong, Wei Zeng, Xianfeng Gu, Paul M. Thompson, Tony F. Chan, and Shing-Tung Yau, *Optimization of surface registrations using beltrami holomorphic flow*, J. Sci. Comput., 50 (2012), p. 557585.

[20] Bruno Lvy, Sylvain Petitjean, Nicolas Ray, and Jrme Maillot, *Least squares conformal maps for automatic texture atlas generation*, in IN SIGGRAPH 02 CONFERENCE PROCEEDINGS, 2002, p. 362371.

[21] C. Wayne Mastin and Joe F. Thompson, *Discrete quasiconformal mappings*, Zeitschrift Fur Angewandte Mathematik Und Physik, 29 (1978), p. 111.

[22] Ulrich Pinkall and Konrad Polthier, *Computing discrete minimal surfaces and their conjugates*, Experimental Mathematics, 2 (1993), p. 1536.

[23] Boris Springborn, Peter Schrder, and Ulrich Pinkall, *Conformal equivalence of triangle meshes*, ACM Trans. Graph., 27 (2008), p. 77:177:11.

[24] W. T. Tutte, *How to draw a graph*, Proc Lond Math Soc, 13 (1963), p. 743767.

[25] Tsz Wai Wong and Hongkai Zhao, *Computation of quasiconformal surface maps using discrete beltrami flow*, UCLA CAM Reports, 12 (2012).

[26] Wei Zeng, Lok Ming Lui, Feng Luo, Tony Fan-Cheong Chan, Shing-Tung Yau, and David Xianfeng Gu, *Computing quasiconformal maps using an auxiliary metric and discrete curvature flow*, Numer. Math., 121 (2012), p. 671703.

[27] Wei Zeng, Lok Ming Lui, Lin Shi, Defeng Wang, Winnie C. W. Chu, Jack C. Y. Cheng, Jing Hua, Shing-Tung Yau, and Xianfeng Gu, *Shape analysis of vestibular systems in adolescent idiopathic scoliosis using geodesic spectra*, in Proceedings of the 13th international conference on Medical image computing and computer-assisted intervention: Part III, MICCAI'10, Berlin, Heidelberg, 2010, Springer-Verlag, p. 538546.