SIAM J. IMAGING SCIENCES
Vol. 12, No. 3, pp. 1392–1424

Computing Quasi-Conformal Folds*

Di Qiu†, Ka-Chun Lam‡, and Lok-Ming Lui†

Abstract. Computing surface folding maps has numerous applications ranging from computer graphics to material design. In this work we propose a novel way of computing surface folding maps via solving a linear PDE. This framework is a generalization of the existing computational quasi-conformal geometry and allows precise control of the geometry of folding. This property comes from a crucial quantity that occurs as the coefficient of the equation, namely, the alternating Beltrami coefficient. This approach also enables us to solve an inverse problem of parametrizing the folded surface given only partial data with known folding topology. Various interesting applications such as fold sculpting on 3 dimensional models, study of Miura-ori patterns, and self-occlusion reasoning are demonstrated to show the effectiveness of our method.

Key words. Beltrami equation, quasi-conformal geometry, mathematical origami, fold modeling

AMS subject classifications. 65D18, 68U05, 65D17

DOI. 10.1137/18M1220042

1. Introduction.

Modeling the folding phenomena of surfaces, as well as the study of its regular patterns such as mathematical origami, has attracted lots of interests in computer graphics, material design, as well as mathematics. Recent examples include origamizing surfaces [8], material design with mathematical origami [10], a FoldSketch system manipulating the folding of clothes [16], modeling curved folding surface used in fabrication and architectural design; see for example [13, 25]. All these approaches treat a folded surface as an embedded surface in the three dimensional space, with the folding being a property of the embedding, rather than something intrinsic to the surface itself.

Very often such an embedding can be nonsmooth, and thus can be harder to deal with directly. For example, a typical origami when viewed as an isometric embedding of some flat surface, is only of C^0 regularity. In other words, the embedding is not differentiable at the folding creases, which however are the key trait of such folded surfaces.

Is it possible to encode the folding as something intrinsic to the surface, in the sense of differential geometry? Since then it would allow us to abstract away some excessive degrees of freedom, such as all possible isometric embeddings of the folded surface when realized in $\mathbb{R}^3$. Such an encoding will have to keep the key properties of the folding, so as to manipulate the folding via the encoding in effective manners.

*Received by the editors October 10, 2018; accepted for publication (in revised form) May 2, 2019; published electronically August 13, 2019.

https://doi.org/10.1137/18M1220042

Funding: The work of the authors was supported by the Hong Kong Research Grants Council GRF project 2130549.

†Department of Mathematics, The Chinese University of Hong Kong, Hong Kong (dqiu@math.cuhk.edu.hk, lmlui@math.cuhk.edu.hk).

‡Department of Computing and Mathematical Science, California Institute of Technology, Caltech, Pasadena, California 91125 (kclam@caltech.edu).

In this paper we propose to use a novel fold encoding scheme that is intrinsic to the surfaces, taking ideas from conformal geometry. We encode folded surface by a "folding map" defined on its encoding domain. The folding map can be determined using a local differential quantity called the *alternating Beltrami coefficient* [22], which takes the form of a complex-valued scalar field defined on the encoding domain. The coefficient represents precisely a more general type of local conformal (i.e., angle) distortion of the folding map, including its possible *change of orientation*, which we use to model folds and its geometry. Our work can be considered as the first application of quasi-conformal methods to the problem of modeling and studying the folding of surfaces.

Computationally, it turns out that conformal distortion data together with very mild constraints can be used to determine the folding map uniquely. Classically, this fact can be seen from the *Riemann mapping theorem*, whose uniqueness part says that the conformal map is uniquely determined once a point of the mapping and the direction of its complex derivative is fixed. In our case, a folding map is the solution of a linear partial differential equation, called the *alternating Beltrami equation* in quasi-conformal theory [22]. We will show in this paper in Proposition 2.13 below, that for a triangulated mesh, pinning as few as two points can determine the folding map uniquely. Intuitively, the coefficients prescribe the deformation of each triangle mesh, in particular, whether it should be flipped or not, and solving the equation is similar to gluing these triangles together. We show that the alternating Beltrami equation admits a quadratic variational formulation, and in fact it generalizes the classical least squares conformal parametrization [15, 9] in the sense that it allows folding with prescribed conformal distortion. Consequently, folding or unfolding can be achieved by simply solving a sparse linear system.

Under this framework, we can pass in-between the folded and unfolded states. By solving this equation on the encoding domain based on the associated coefficients, we can fold a domain into the corresponding folded surface. The inverse of the folding map can also be computed using the equation of the same type defined on the folded surface, which unfolds it back to its encoding domain. What is more, the encoding scheme with its associated folding or unfolding maps also allows us to relate different folded surfaces with the same "folding topology" via deformation. With this we can formulate optimization problems to find the "best" positioning of the folding creases under certain criteria by deforming an existing one. We use this idea to solve the problem of recovering a *flat-foldable* surface with self-occlusion. In other words, we show that it is possible in many cases to use only the topology and some partial geometric information of the folding to reconstruct the entire flat-foldable surface. Intuitively, our approach is based on a factorization of a quasi-conformal folding, in a spirit similar to the classical Stoilow factorization [1]. That is, we factor a quasi-conformal folding through a quasi-conformal deformation of the domain, followed by a flat-foldable folding map. Moreover, if the given domain's geometry is close to the true one that corresponds to the observed folded surface, then the two factored mappings should have low conformal distortion. This important observation leads us to a fixed-point-like algorithm that reconstructs both the domain and the folded surface in alternating steps.

The paper is organized as follows. In section 2.1 we develop the topological notion of "folding homeomorphism" and the associated singular set configuration, which are formal definitions of the folding map and folding topology mentioned previously. We then explain

the geometric meaning of the alternating Beltrami coefficient and the associated equation, visualize the folding effect, and give the definition of flat foldability. In section 2.2 we develop the variational formulation and give its discrete geometric interpretation, together with its discretization and implementation. In section 3 we define the reconstruction problem of flat-foldable surfaces with self-occlusion, and develop the "reinforcement iteration" algorithm to solve it. Finally, in sections 4 and 5 we demonstrate the results of the algorithm and various interesting applications including generating new flat-foldable *Miura-ori* prototypes, foldlike texture generation, fold and cusp sculpting on surfaces, fold in-painting for surfaces, and occlusion reasoning for flat-foldable surfaces.

1.1. Related work. Here we briefly list some important related works in this area, while they are by no means an exhaustive survey.

Computational quasi-conformal geometry. Computational technique of conformal map [15, 9] turned out to be very useful in computer graphics. Since the seminal work of Gu et al. [11], the conformal geometry framework in surface processing tasks has advanced significantly. Earlier work generalizing these ideas is already implicit in the work of Zayer, Rossl, and Seidel [24]. The quasi-conformal extension of the surface registration framework was proposed by Lui and his coauthors [18, 17, 5], with successful applications to medical image registration and surface registration. The quasi-conformal method is able to handle large deformations, where conformal methods typically fail. Our work is a generalization along this line, which allows the manipulation of folding, and opens up a new research area to explore.

Modeling surface folding and mathematical origamis. In computer graphics there has been a notable amount of work on modeling the folding phenomena of surfaces. Many interesting works focus on 3 dimensional (3D) interactive design. These include the method of thin plate form with explicit user control of folding angles for interactive 3D graphics design in [25], which can also achieve the sharp folding edges as we do here. Our framework and techniques are completely different, especially here we are taking advantage of the fact that alternating Beltrami equation can be solved effectively in 2 dimensions. On the other hand, there are studies of developable surface design with curved folding [13], taking advantage of the special quad meshes, while we don't have this restriction, but the focus and techniques are rather different. We must also mention the work of Demaine and Tachi [8], who developed algorithms to fold a planar paper into arbitrary 3D shapes. The study of the folding phenomena also has industrial applications such as the 4 dimensional printing of [14] and material design in [10]. In comparison with the work above, we encode the folding and its geometry in an intrinsic way, and hence 3D geometric features of the folded surface, such as mean curvature, do not belong to our framework but can be tackled by pre- or post-processing. We expect to discover interesting connections with others in future work.

2. Computing quasi-conformal folds.

2.1. Definitions of folding homeomorphism, singular set configuration, and derivation of alternating Beltrami equation.

2.1.1. Topological description. The notion of folding used in this paper departs from those of a 3D nature, as in [13]. Instead, we model it via the the continuous map from the domain to the target with designated change of orientation. This is made precise in the

following definitions. For simplicity, we focus on the situation where the domain and target are simply connected subsets of $\mathbb{R}^2 \cong \mathbb{C}$. The extension to multiply connected domains is straightforward.

Definition 2.1 (two color map). *Let X be a simply connected subset of $\mathbb{R}^2$ and Σ be a union of finite number of curves in X. A pair (X, Σ) is said to be a two color map if $X \setminus \Sigma$ is a disjoint union of surfaces whose boundary belongs to Σ, and each member of the union is assigned uniquely with one of the two distinct colors such that the closures of any two distinct members assigned with the same color intersect at most at a finite number of points.*

Definition 2.2 (folding homeomorphism and its singular set configuration). *Let X and Y be oriented, simply connected subsets of $\mathbb{R}^2$. A continuous, surjective, and discrete map $f : X \to Y$ (i.e., $f^{-1}(y)$ are isolated in X for all $y \in Y$) is called a folding homeomorphism if there is a subset $\Sigma \subset X$ with (X, Σ) forming a white-black two color map, such that when restricting to the white (or black) region, f is an orientation-preserving (or -reversing) homeomorphism. The two color map (X, Σ) is called the singular set configuration of f, or simply referred to as Σ if clear from the context. The white and black regions are denoted by X^+, X^-, respectively.*

Note that we have required the map to be discrete in order to avoid the case of degeneracy.

For convenience we would also like to give names to the points in the singular set according to the properties of f. The definition below will classify the usually encountered situation and suffices for our purposes in this paper. Readers can also refer to Gutlyanskii et al. [12] for related materials.

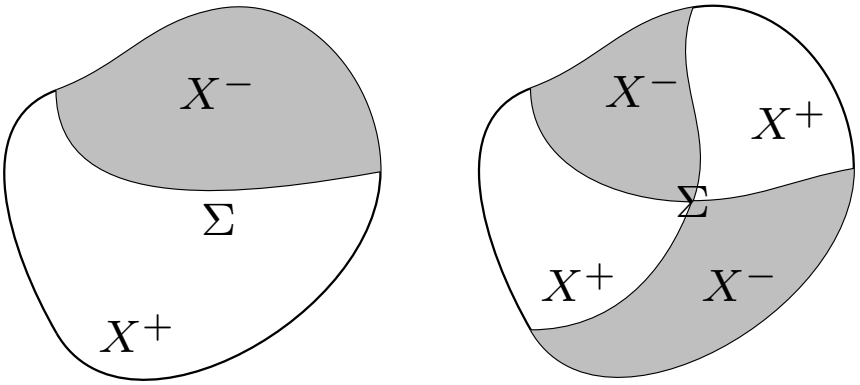
Definition 2.3 (folding point and cusp point). *Let $f : X \to Y$ be a folding homeomorphism with singular set configuration Σ.*

- *A point $p \in \Sigma$ is called a folding point if there is an open neighborhood U of p such that $U \setminus \Sigma$ is disconnected into exactly 2 simply connected components, and the restriction $f\big|_U$ after precompositing with a suitable homeomorphism $\varphi : \tilde{U} \to U$ can be written as $\tilde{U} \ni (x, y) \mapsto (x, |y|)$, where $\varphi^{-1}(U \cap \Sigma)$ plays the role of x-axis.*
- *A point $p \in \Sigma$ is called a cusp point if there is an open neighborhood U of p such that $U \setminus \Sigma$ is disconnected into exactly $2n$ simply connected components, $n > 1$, and the remaining points $p' \in (\Sigma \cap U) \setminus \{p\}$ are all folding points.*
- *The collection of paths in Σ consisting of all folding points is called folding lines.*

We illustrate these concepts in Figure 1. For a simple explicit example, consider the map

$$f(x+iy) = \begin{cases} x+iy & \text{if } ax+by \geq 0, \\ x+iy-\frac{2(ax+by)(a+bi)}{a^2+b^2} & \text{if } ax+by < 0, \end{cases}$$

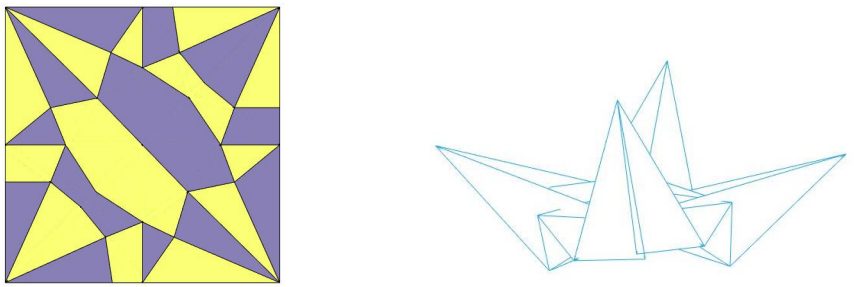
where $a > 0, b \in \mathbb{R}$. Then we see that f folds the half-plane $\{x+iy : ax+by < 0\}$ into the other half $\{x+iy : ax+by > 0\}$. Explicit examples of a cusp are more involved to write down, but can be found in [12]. A more famous example is the paper crane origami, whose singular set configuration is shown in Figure 2, where for better visualization we use yellow and purple instead of white and black. The paper crane origami is in fact flat foldable, which we will define in Definition 2.4.



(a) The singular configuration contains only folding points: the folding line separates X^- and X^+.

(b) The singular configuration contains both folding points and a single cusp point: the cusp point joins the folding lines.

Figure 1. *Illustration of Definition 2.3.*



(a) Singular set configuration of the paper crane.

(b) The paper crane obtained by solving an alternating Beltrami equation.

Figure 2. *The paper crane origami.*

Since change of orientation necessarily creates nonbijectivity, the inverse of a folding homeomorphism f is no longer a function. Nevertheless it is still possible to define an "unfolding homeomorphism" that unfolds a folded surface once we can distinguish the image of the points in $f^{-1}(p)$. Formally, one can consider the map

$$F : (p, p) \mapsto (p, f(p)), \ p \in \Omega,$$

where f is a folding homeomorphism. F is bijective, hence has an inverse that unfolds the folded surface. In practice, if the fold is represented as a mesh as in Figure 2(b), the points in the image of the folding map are naturally distinguished by the mesh data structure. Locally, the unfolding homeomorphism undoes the change of orientation done by the corresponding

folding homeomorphism. As we shall see below in Remark 2.16, since we are dealing with a differential equation, we need only to know on which triangle the orientation is reversed, and thus need not worry about the nonbbijectivity.

So far the above discussions have been topological. In order to compute the folding map, we need to find geometric quantities that encode the folding geometry and methods that reconstruct the folding from these quantities. As pointed out in the introduction, our encoding scheme is based on intrinsic differential geometry, namely, it does not depend on how the folded surface is embedded in $\mathbb{R}^3$. At the same time, the encoding can be seen as a variable to be inferred from optimization problems, so that we can start with an initial encoding domain and solve the optimization problem to get the optimal one. We shall turn to the geometric quantity and method of reconstruction next.

2.1.2. The quasi-conformal geometry of folding homeomorphisms. We can equip the domains with Riemannian metrics and thus talk about the intrinsic geometry of the folding homeomorphism $f : (X, g_1) \to (Y, g_2)$. To measure the local distortion of f is to compare the pullback metric $f^* g_2$ with the original metric g_1. In the other direction, the knowledge of the distortion will give rise to a PDE that characterizes the map. We shall describe this point in details below.

For our purpose, let us take an open set $\Omega \subset \mathbb{R}^2 \cong \mathbb{C}$ in X which does not contain points in Σ, and put g_2 to be the Euclidean metric. If we consider the pullback metric by the unknown map f as given data in the form of a matrix field $H : \Omega \to \mathbf{S}_{++}$, where $\mathbf{S}_{++}$ denotes the space of symmetric positive definite 2×2 matrices, and assume f is differentiable, it then satisfies the nonlinear system [1]

$$Df(z)^T Df(z) = H(z) \ \forall z \in \Omega,$$

where $f = (u, v)^T, z = (x, y)^T, Df(z) = \left[\begin{smallmatrix} u_x & u_y \\ v_x & v_y \end{smallmatrix}\right]$. Surprisingly enough, it is possible to reduce the above to a linear equation if the data are given up to multiplying an everywhere positive function. This is the essential advantage for us to introduce two dimensional conformal geometry to our problem. To do this, denote

$$\mathbf{S}(2) = \{M \in \mathbf{S}_{++} : \det M = 1\}.$$

And let $Q : \Omega \to S(2)$ be such that $H(z) = \phi(z)Q(z)$, so the above nonlinear system is expressed as

$$Df(z)^T Df(z) = \phi(z)Q(z), \ \phi(z) > 0.$$

By taking determinants on both sides, we get

$$\phi(z) = |J_f(z)|,$$

where $J_f(z) = \det Df(z)$. Note that the absolute value is necessary since f may be orientation reversing. As a result, we get

$$Df(z)^T Df(z) = |J_f(z)|Q(z).$$

Multiplying $Df(z)^{-1}$ on the right of both sides, and write

$$f = (u, v)^T, Q = \begin{bmatrix} q_{11} & q_{12} \\ q_{12} & q_{22} \end{bmatrix},$$

we obtain the system

$$\begin{bmatrix} u_x & u_y \\ v_x & v_y \end{bmatrix}^T = \operatorname{sgn}(J_f(x)) \cdot \begin{bmatrix} q_{11} & q_{12} \\ q_{12} & q_{22} \end{bmatrix} \begin{bmatrix} v_y & -u_y \\ -v_x & u_x \end{bmatrix}.$$

It is a straightforward matter to rewrite the above system in complex derivatives, obtaining

$$\frac{\partial f}{\partial \bar{z}}(z) = \mu(z) \frac{\partial f}{\partial z}(z),$$

where $\mu = \frac{q_{11}-q_{22}+2iq_{12}}{q_{11}+q_{22}+2\cdot\operatorname{sgn}(J_f(x))}$, $\frac{\partial f}{\partial \bar{z}} = (u_x - v_y) + i(u_y + v_x)$, and $\frac{\partial f}{\partial z} = (u_x + v_y) + i(-u_y + v_x)$. This is called the alternating Beltrami equation, coined by Srebro and Yakubov [22]. The name refers to the fact that

$$\begin{cases} |\mu| < 1 & \text{if } \operatorname{sgn}(J_f(x)) > 0, \\ |\mu| > 1 & \text{if } \operatorname{sgn}(J_f(x)) < 0, \end{cases}$$

and hence it differs from the classical Beltrami equation in that the modulus of the coefficient is only required to be bounded away from 1. This motivates a more analytic definition for the folding homeomorphism, which is the principal mathematical subject of this paper.

Definition 2.4 (quasi-conformal map with folds). *A folding homeomorphism $f : \Omega \subset \mathbb{C} \to \mathbb{C}$, $K \geq 1$, is called a generalized K-quasi-conformal map with singular configuration Σ if it is a solution of an alternating Beltrami equation $\frac{\partial f}{\partial \bar{z}}(z) = \mu(z)\frac{\partial f}{\partial z}(z)$, such that in Ω^+ and Ω^- it holds that $|\mu(z)| < 1$ and $|\mu(z)| > 1$, respectively, and moreover satisfies the bound*

$$\left|\frac{1 + |\mu(z)|}{1 - |\mu(z)|}\right| \leq K$$

for all $z \in \Omega$ except for a set of Lebesgue measure zero. In particular, the case $K = 1$ will be called generalized conformal, or flat foldable, or a planar origami. Note that generalized conformality corresponds to $\mu = 0$ or ∞.

The quotient inside the bound has the interpretation of a linear distortion of the map; see [1]. The above definition of flat foldability is adapted to our computational problem. It encompasses the case where a surface is rigidly flat folded, whose folding lines of the singular set configuration are all Euclidean geodesics.

To get a better picture of the alternating Beltrami equations, we illustrate it with the effect of the Beltrami coefficients on a single triangle (i.e., the linearized effect at the tangent space level). This should provide one with geometric intuition for the solutions on a triangulated mesh.

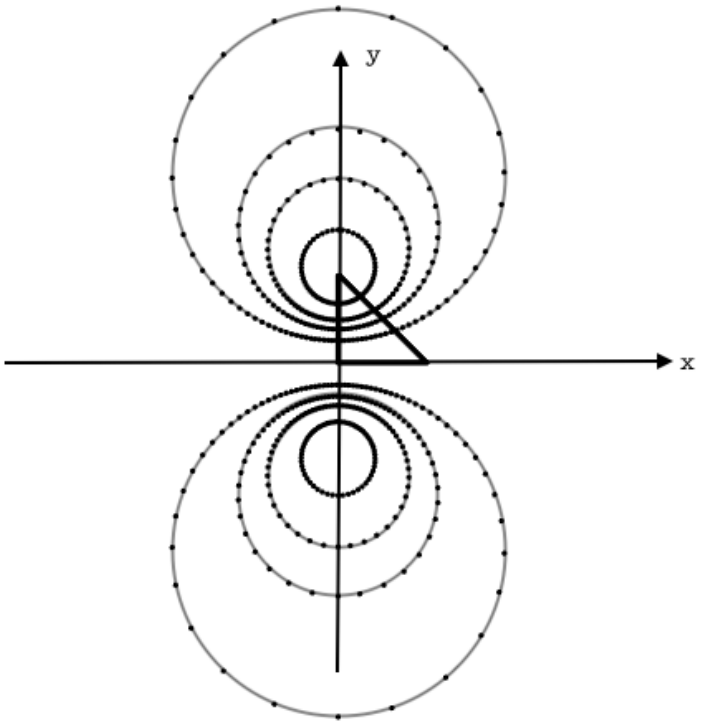


Figure 3. *The trajectory of the third vertex under different values of Beltrami coefficients. Circles represent the situation when* $|\mu| = 1/5, 7/20, 9/20, 3/5, 5/3, 20/9, 20/7, 5$ *respectively.*

Let us rewrite the Beltrami equation as a system of first order PDEs in the usual Cartesian coordinate. Suppose $f : (x, y) \mapsto (u, v)$ satisfies the equation $\frac{\partial f}{\partial \bar{z}}(z) = \mu(z)\frac{\partial f}{\partial z}(z)$. If we write $\mu = \rho + i\tau$, then it's not hard to see that

$$\begin{bmatrix} u_y \\ v_y \end{bmatrix} = \frac{1}{(1+\rho)^2 + \tau^2} \begin{bmatrix} 2\tau & |\mu|^2 - 1 \\ 1 - |\mu|^2 & 2\tau \end{bmatrix} \begin{bmatrix} u_x \\ v_x \end{bmatrix}; \tag{1}$$

here we have assumed $\rho \neq -1$ and $\tau \neq 0$.

Hence for a single triangle, on which we assume f is linear, the map is determined up to a similarity transform (uniform scaling and rotation) in the target domain. Now suppose f maps a domain triangle $[v_1, v_2, v_3] = [(0, 0), (1, 0), (x, y)]$ to the target triangle $[w_1, w_2, w_3] = [(0, 0), (1, 0), (u(x, y), v(x, y))]$. Then

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} 1 & \frac{2\tau}{(1+\rho)^2+\tau^2} \\ 0 & \frac{1-|\mu|^2}{(1+\rho)^2+\tau^2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \tag{2}$$

One can check that the set of points for the family of μ with each fixed modulus $|\mu| \neq 1$ form a circle, whereas in the case $|\mu| = 1$ the circle degenerates to the x-axis. An illustration of this fact is shown in Figure 3.

As one goes beyond to the case $|\mu| > 1$, where the antidiagonal terms experience a change of sign, this leads to a "flipping" of the triangle. In fact, for a single triangle, everything remains the same after a mirror reflection about the x-axis, and the case $\mu = \infty$ corresponds to the *anticonformality* of the map. Here ∞ should be understood as the infinity point in the Riemann sphere. What is more, for each fixed argument $\arg(\mu)$ and let the modulus $|\mu|$ vary, the set of solution points form an arc of a circle, passing through the points (x, y) and $(x, -y)$. Altogether, we see that the Beltrami coefficients in effect form a bipolar coordinate in the plane containing the target triangle. Therefore, it describes all possible angular distortions at the tangent space level, including those having a change of orientation.

2.2. Energy formulation. In this section we turn to the computational methods to solve the Beltrami equation. Previously proposed methods include the *Beltrami holomorphic flow method* as in [18], or the decoupling method as in [17]. Both require entire boundary information for solving the Beltrami equation and it can be unrealistic in applications. Fortunately, it turns out to be also unnecessary once we realize the coupling of the two coordinate functions of the map. This coupling arises naturally in an energy functional of least squares type. For completeness, we analyze this problem below since we did not find it in the literature.

The formulation here takes inspiration from the well-known *least squares conformal energy*, studied in [15, 9], which takes into account the coupling of u and v. Its continuous formulation is

$$\int_\Omega \|\nabla u + \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \nabla v\|^2 dxdy.$$

The corresponding matrix associated with its discretization is the well-known *cotangent weight* matrix minus a certain "area matrix" [19, 9]. This area matrix in fact plays the role of certain Neumann boundary conditions. One would expect analogous results to hold in the quasi-conformal setting.

But in the quasi-conformal case, it is not an entirely trivial matter to formulate the correct analog. It turns out the energy must give rise to a pair of second order elliptic equations as a necessary condition.

2.2.1. The decoupling method and the necessary condition. Perhaps the most straightforward way to solve the Beltrami equation is to decouple the corresponding first order system into two independent second order equations, namely, the following.

Proposition 2.5 (necessary condition). *Suppose for $z \in \Omega \setminus \Sigma$, $f(z) = u(z) + iv(z)$ satisfies the equation $\frac{\partial f}{\partial \bar{z}}(z) = \mu(z)\frac{\partial f}{\partial z}(z)$. Assume the domain Ω is given the usual Euclidean geometry, and $|\mu| \neq 1$, then we have*

$$\begin{cases} -\nabla \cdot (A \nabla u(z)) = 0, \\ -\nabla \cdot (A \nabla v(z)) = 0, \end{cases} \tag{3}$$

where

$$A = \frac{1}{1 - |\mu|^2} \begin{bmatrix} (\rho - 1)^2 + \tau^2 & -2\tau \\ -2\tau & (1 + \rho)^2 + \tau^2 \end{bmatrix} \quad \text{and} \quad \mu = \rho + i\tau.$$

Proof. Observe that $\frac{\partial f}{\partial \bar{z}}(z) = \mu(z)\frac{\partial f}{\partial z}(z)$ can be transformed into

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} A \begin{bmatrix} v_x \\ v_y \end{bmatrix}.$$

Then making use of the commutativity of second order partial derivatives $u_{xy} = u_{yx}$ under the Euclidean coordinate, we obtain

$$\nabla \cdot (A \nabla u(z)) = 0.$$

The other equation is obtained in a similar way. ∎

Remark 2.6. Note that the coefficient matrix A is positive (or negative) definite if $|\mu| < 1$ (or $|\mu| > 1$, respectively). If U is any open neighborhood, on which A is either positive or negative but not both, then it is not hard to see that system (3) is the Euler–Lagrange equations of the Dirichlet-type energies

$$E_{\tilde{A}}(u; U) = \frac{1}{2}\int_U \|\tilde{A}^{1/2}\nabla u\|^2 dxdy, \quad E_{\tilde{A}}(v; U) = \frac{1}{2}\int_U \|\tilde{A}^{1/2}\nabla v\|^2 dxdy \tag{4}$$

with Dirichlet boundary conditions, where $\tilde{A}$ denotes A if A is positive definite, or $-A$ if A is negative definite. Therefore, we see that in general the global variational problem must be separated according to whether $|\mu| < 1$ or $|\mu| > 1$ in the domain Ω. We shall often denote Ω^+ (or Ω^-) as the largest open subset such that $|\mu| < 1$ (or $|\mu| > 1$, respectively), which is consistent with the previous notation.

The derived system (3) is a *necessary condition* that in principle should be satisfied by any other method which solves the equation in the Euclidean domain Ω. This motivates the following.

Definition 2.7. *Let $\mu = \rho + i\tau$ be a complex-valued function defined on the domain Ω. The* least squares quasi-conformal energy *of the map $z = (x, y) \mapsto (u, v)$* against *the Beltrami coefficient μ is defined to be*

$$E_{LSQC}(u, v; \mu) = \frac{1}{2}\int_\Omega \|P\nabla u + JP\nabla v\|^2 \, dxdy,$$

where

$$P = \frac{1}{\sqrt{1-|\mu|^2}}\begin{bmatrix} 1-\rho & -\tau \\ -\tau & 1+\rho \end{bmatrix}, \; J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},$$

so that $P^T P = A$ as in (3). Note that if $|\mu| > 1$ we allow entries of P to be imaginary, and $\|\cdot\|^2 = \langle \cdot, \cdot \rangle$ remains as the usual Euclidean norm (not Hermitian).

Let's first consider the uniform case and assume that $|\mu| < 1$. The case of $|\mu| > 1$ defers by a negative sign. Thanks to the important observation that $P^T J P = J$, we have the following identity

$$E_{LSQC}(u, v; \mu) = (E_A(u; \Omega) + E_A(v; \Omega)) - \mathcal{A}(u, v),$$

where A is the same matrix described previously in (3), and

$$\mathcal{A}(u,v) := \int_\Omega (u_y v_x - u_x v_y)\, dxdy$$

is the area of the target surface.

Remark 2.8. Assume that $|\mu| < 1$. Observe that we have obtained the analog of the classical lower bound of the Dirichlet energy

$$E_A(u) + E_A(v) \geq \mathcal{A}(u,v). \tag{5}$$

This simply follows from the fact that $E_{LSQC}(u,v;\mu) \geq 0$. The vanishing of this energy is equivalent to the existence of $f = u + iv$ as a solution of the Beltrami equation with coefficient μ. The existence is guaranteed for measurable Beltrami coefficients μ with $\|\mu\|_{L^\infty(\Omega)} < 1$, known as the *measurable Riemann map theorem*. Note also that the solution of the Beltrami equation is unique up to postcomposition of conformal maps [1].

If we assume the domain $\partial\Omega$ has a Lipschitz boundary, then the quantity $\mathcal{A}(u,v)$ is equal to the following integral on the boundary,

$$\frac{1}{2}\int_{\partial\Omega} (v\nabla u - u\nabla v) \times \nu\, d\Gamma,$$

where $\nu(z)$ is the outer unit normal vector, and $d\Gamma$ is the standard measure of $\partial\Omega$. Actually, the coupling between u and v is realized as a certain boundary condition applied to solving (3). The following derivation of the second order equations with boundary condition is standard.

Theorem 2.9. *Suppose μ is uniformly bounded away from 1, Ω is connected with a Lipschitz boundary, and suppose there exists one pair (u,v), $u,v \in W^{2,2}(\Omega)$, such that*

$$E_{LSQC}(u,v;\mu) = \arg\inf_{\tilde{u},\tilde{v}\in W^{1,2}} E_{LSQC}(\tilde{u},\tilde{v};\mu),$$

then they satisfy the following Neumann boundary problem:

$$\begin{cases} -\nabla\cdot(A\nabla u) = 0 \ \text{in } \Omega, \\ -\nabla\cdot(A\nabla v) = 0 \ \text{in } \Omega, \\ \partial_{A\nu} u + \nabla v \times \nu = 0 \ \text{on } \partial\Omega, \\ \partial_{A\nu} v - \nabla u \times \nu = 0 \ \text{on } \partial\Omega, \end{cases} \tag{6}$$

where as before $\nu(z)$ is the outer unit normal vector.

2.2.2. Generalized quasi-conformal energy. Because of the change of orientation, the energy formulation and the associated system of equations has to be accordingly modified. It is then crucial to study the interaction between regions of the domain that corresponds to different orientations of f.

First of all, it follows from arguments in Remarks 2.6 and 2.8 that the alternating Beltrami equation, when *restricted to regions of constant orientation*, is equivalent to vanishing of the "energies"

$$E^+_{LSQC}(u,v;\mu) := \frac{1}{2}\int_{\Omega^+} \|P\nabla u + JP\nabla v\|^2\, dxdy = 0,$$
$$E^-_{LSQC}(u,v;\mu) := \frac{1}{2}\int_{\Omega^-} \|P\nabla u + JP\nabla v\|^2\, dxdy = 0,$$

where $\Omega^+ = \text{int}\,\{z \in \Omega : |\mu(z)| < 1\}$, $\Omega^- = \text{int}\,\{z \in \Omega : |\mu(z)| > 1\}$. Recall that when $|\mu| > 1$ we allow entries of P to be imaginary, hence, $E^-_{LSQC}(u,v;\mu)$ is in fact negative definite.

To obtain the global solution, one could solve the equation individually in Ω^+ and Ω^- and glue the solution along the singular set configuration. It turns out that this can be done implicitly. The problem now is how to combine the quasi-conformal energies on regions with different orientations into a single "energy," so that we can solve the alternating Beltrami equation on the entire domain in one shot.

Theorem 2.10 (generalized quasi-conformal energy). *Assume there are only finitely many cusp points. Define the generalized quasi-conformal energy with Beltrami coefficient μ of the map $z = (x,y) \mapsto (u,v)$ in $W^{2,2}$ to be*

$$E_{GQC}(u,v;\mu) = E^+_{LSQC}(u,v;\mu) - E^-_{LSQC}(u,v;\mu).$$

Then the alternating Beltrami equation with Beltrami coefficient μ is a critical point of the above energy.

Proof. By taking a test function in the interior of constant orientation or near the boundary $\partial\Omega$, the critical point property in these regions is verified no differently from the classical case. It now suffices to check the critical point property for the region near the singular set configuration. Since the number of cusp points is finite, it will not contribute to the integration on the singular set. Hence it suffices to work locally in a small neighborhood U that contains only folding points, like the situation in Figure 1(a). Take any smooth test function ϕ compactly supported in U. Then by setting

$$\frac{d}{d\epsilon}\Big|_{\epsilon=0} E_{GQC}(u + \epsilon\phi, v) = 0$$

we obtain

$$\left(\int_{\Omega^+} - \int_{\Omega^-}\right) \langle P\nabla\phi, P\nabla u\rangle + \langle P\nabla\phi, JP\nabla v\rangle\, dxdy = 0.$$

Integrating by parts, and repeating the same steps for v, we can derive the following Euler–Lagrange system,

$$\begin{cases} -\nabla\cdot(\tilde{A}\nabla u) = 0 \text{ in } \Omega^+\cup\Omega^-, \\ -\nabla\cdot(\tilde{A}\nabla v) = 0 \text{ in } \Omega^+\cup\Omega^-, \\ \partial_{\tilde{A}\nu} u + \nabla v \times \nu = 0 \text{ on } \partial\Omega\cup\Sigma, \\ \partial_{\tilde{A}\nu} v - \nabla u \times \nu = 0 \text{ on } \partial\Omega\cup\Sigma, \end{cases} \tag{7}$$

where

$$A = \frac{1}{1-|\mu|^2}\begin{bmatrix}(\rho-1)^2+\tau^2 & -2\tau \\ -2\tau & (1+\rho)^2+\tau^2\end{bmatrix},$$

$\tilde{A} = A$ on Ω^+ and $\tilde{A} = -A$ on Ω^-, $\nu = \nu(z)$ is the outer unit normal vector when $z \in \partial\Omega^+$, or equivalently the inner unit normal if we regard $z \in \partial\Omega^-$. Note that the second order equation outside of the singular set and boundary has the same form, and corresponds to the alternating Beltrami equation with coefficient μ. Most importantly, the boundary condition matches up from both sides of the folding line. This finishes the proof. ∎

Remark 2.11. Note that the above Neumann boundary problem is somewhat different from convention since the singular set lies in the interior but is treated like boundary. But this very condition can be seen as the way to glue two pieces of solutions on Ω^+ and Ω^- together along the singular set configuration.

2.2.3. Discretization and implementation details. First we discuss the case of least squares quasi-conformal energy and later extend it to the generalized case. We discretize the the equation (6) on a linear triangular mesh $\mathcal{T}$, which is encoded as a list of vertices V and a list of triangles $\mathcal{T}$ (by a mild abuse of notation) taking indices into V. We denote the number of vertices by $|V|$ and number of triangles by $|\mathcal{T}|$. The second order operator $\nabla\cdot(A\nabla)$ is a variant of the Laplacian. Its discretization amounts to expressing the following sum

$$\sum_{T\in\mathcal{T}} \langle P\nabla\varphi(T), P\nabla\phi(T)\rangle_T \tag{8}$$

for any test functions φ, ϕ defined on the vertices V into a quadratic form $\varphi^T\mathcal{L}_\mu\phi$. Here, $\langle\cdot,\cdot\rangle_T$ is the 2D Euclidean inner product scaled by the area of the triangle T. On an oriented triangle $T = [v_0, v_1, v_2]$, since the functions being considered are linear on triangles, the gradient of a function $\varphi = (\varphi_0, \varphi_1, \varphi_2)$ on this triangle is given by

$$\nabla\varphi = \frac{1}{2\mathrm{Area}(T)}\begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix}\sum_{i=0,1,2}\varphi_i(v_{2+i}-v_{1+i}).$$

where indexing modulo 3 as appropriate. Observe that

$$\begin{bmatrix}1-\rho & -\tau\\ -\tau & 1+\rho\end{bmatrix}\begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix} = \begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix}\begin{bmatrix}1+\rho & \tau\\ \tau & 1-\rho\end{bmatrix}$$

Hence, denoting

$$v_i' = P^{-1}v_i,$$

we have

$$P\nabla\varphi = \frac{1}{2\mathrm{Area}(T')}\begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix}\sum_{i=0,1,2}\varphi_i(v_{2+i}'-v_{1+i}').$$

Therefore, denoting the triangle $T' = [v_0', v_1', v_2']$,

$$\begin{aligned}\langle P\nabla\varphi(T), P\nabla\phi(T)\rangle_T &= -\frac{1}{4\mathrm{Area}(T')}\sum_{i,j}\varphi_i\phi_j(v_{2+i}'-v_{1+i}')^T(v_{2+j}'-v_{1+j}')\\ &= -\sum_{i,j}\omega_{ij}(T)\varphi_i\phi_j\end{aligned}$$

where

$$\omega_{ij}(T) = \begin{cases} -\frac{1}{2}\cot\theta'_k,\ k \neq i,j & \text{if } i \neq j, \\ \frac{1}{2}(\cot\theta'_{i+1} + \cot\theta'_{i+2}) & \text{if } i = j, \end{cases}$$

where θ'_k is the angle of at the vertex v'_k. This is noting but the cotangent weight with angles changed by the effect of μ.

The expression for the area integral $\mathcal{A}(u,v)$ is unchanged from the least squares conformal case [15], and hence we have the following.

Corollary 2.12. *The quadratic form (up to a nonzero constant scaling) associated with the triangular mesh discretization of the least squares quasi-conformal energy is given by the following* $2|V| \times 2|V|$ *matrix*

$$M := diag(\mathcal{L}_\mu, \mathcal{L}_\mu) - 2\mathcal{A},$$

which is applied to the $2|V|$*-coordinate vector* $\mathbf{x} = (u,v)$*. The discrete version of* (6) *is then* $M\mathbf{x} = \mathbf{0}$*. Here* $\mathcal{L}_\mu$ *is the cotangent matrix associated with the operator* $\nabla \cdot A\nabla$ *and* $\mathcal{A}$ *is the area matrix of the target triangular mesh.*

Proof. It follows from the discussion above that the $|V| \times |V|$ matrix $\mathcal{L}_\mu$ corresponds to the discretization of the differential operator $\nabla \cdot (A\nabla)$. The area matrix has nonzero entries only corresponding to the boundary vertices. It is then immediate to check, by examining the corresponding rows of the linear system, that for interior vertices, the solution (u,v) satisfies $-\nabla \cdot (A\nabla u) = 0$ and $-\nabla \cdot (A\nabla v) = 0$, while on the boundary, it satisfies $\partial_{A\nu} u + \nabla v \times \nu = 0$ and $\partial_{A\nu} v - \nabla u \times \nu = 0$, respectively. ∎

In order to obtain a nontrivial solution to the system $M\mathbf{x} = 0$, it turns out one need only pin down at least two vertices. The precise statement is contained in the following proposition.

Proposition 2.13. *Suppose* $|\mu|$ *is uniformly bounded away from* 1*, and the triangular mesh is connected and has no dangling triangles (i.e., there are no triangles which share a common vertex but no common edge). Let* I_{pin} *be the indices of the points to be pinned, with cadinality* $|I_{pin}| \geq 2$*,* I_{free} *be the indices for the free points, and* M *be the matrix defined as in Corollary* 2.12*. Then the* $2|I_{free}| \times 2|I_{free}|$ *submatrix* M_{free} *of* M *indexed by the free points has full rank.*

Proof. The idea is essentially identical to the proof in [15] and we only sketch the main argument and the modification needed here. The key observation is that the (topological) triangular mesh satisfying our assumption can be constructed incrementally using two operations:

- glue: adding a new vertex and connecting it to two neighboring vertices;
- join: joining two existing vertices.

The proof is based on this observation. To wit, we express E_{LSQC} as $\|B\mathbf{x} - \mathbf{b}\|^2$, where B is of size $2|\mathcal{T}| \times 2|I_{free}|$. It then suffices to prove B has full rank. One then proceeds by proving the incremental construction preserves the full rank property. Since the modulus of the Beltrami coefficient associated with the new triangle is bounded away from 1, the associated matrix coefficients are nonzero real numbers. And therefore the same argument in [15] applies. ∎

Remark 2.14. We now have a geometric understanding in the discrete case. In fact, it can be regarded as a conformal map with the domain given a different *conformal structure*. This viewpoint in fact has already been demonstrated previously when we derived the Beltrami

equation. In the case of a triangular mesh, this structure can be thought of as the assignment of the angle triples $(\alpha_1, \alpha_2, \alpha_3)$ to each triangular face of the mesh or, equivalently, the associated Beltrami coefficient. Under this viewpoint we can relate many algorithms from conformal geometric processing to their quasi-conformal counterparts.

Remark 2.15. It is usually a preferred practice to choose these two points far apart from each other to reduce excessive local scale change, as is the case in the conformal flattening task [15]. This is because the triangle angles associated with the Beltrami coefficients as given may not be strictly realizable as a planar mesh, which is in contrast to the continuous case where Beltrami coefficients, under very mild conditions [1], admit solutions as the zeros of the associated least squares quasi-conformal energies. Hence given arbitrary Beltrami coefficients on a triangular mesh, the above discrete system in general yields a minimizer whose energy is small but nonzero. Similarly, if the constraint contains more than two points, the system becomes overdetermined and therefore may not be compatible with the given Beltrami coefficients and, consequently, the solution may have nonzero energy against the given Beltrami coefficients.

The implementation for the generalized quasi-conformal energy is essentially identical to the previous case, except for the separate treatments for the oriented preserving and reversing parts of the domain. This means when assembling the matrix using triangle elements as in (8), for the orientation preserving elements it is identical to the previous case, while for the orientation reversing elements we need to multiply by a minus sign.

Remark 2.16. An unfolding map also satisfies an alternating Beltrami equation, thus it can be computed in the same way, since in order to assemble the matrix M we need only to know the coefficients on each triangular mesh element. In fact, the Beltrami coefficient for the inverse map at a corresponding location can be verified to be the same using a composition formula [1] which is local in nature. The computation of the unfolding map will be important in the reinforcement iteration introduced later.

In the following, solving (alternating) Beltrami equations using the described method will be denoted as

$$\mathcal{T}' = \mathrm{LSQC}(\mathcal{T}, (\mu_T)_{T\in\mathcal{T}}, \mathcal{C}),$$

where $\mathcal{T}'$, $\mathcal{T}$ are the computed target triangular mesh and the domain mesh, respectively; $(\mu_T)_{T\in\mathcal{T}}$ is the set of Beltrami coefficients on each face, and $\mathcal{C}$ is the set of constraints.[1]

3. Recovering a flat-foldable surface from self-occlusion and the reinforcement iteration.

When dealing with flat-foldable surfaces, one of the fundamental problems is that the geometric details are often obscured from *self-occlusions*. If physical unfolding is not possible, we then have an *inverse problem* of inferring the geometry from the visible portion of the surface. In this paper we will make the following qualitative assumption about what kind of data are available to solve the inverse problem of inferring the precise geometry of a partially given flat-foldable surface:

[1]The source code is available at https://github.com/sylqiu/Least-square-beltrami-solver, and it supports the alternating Beltrami equation.

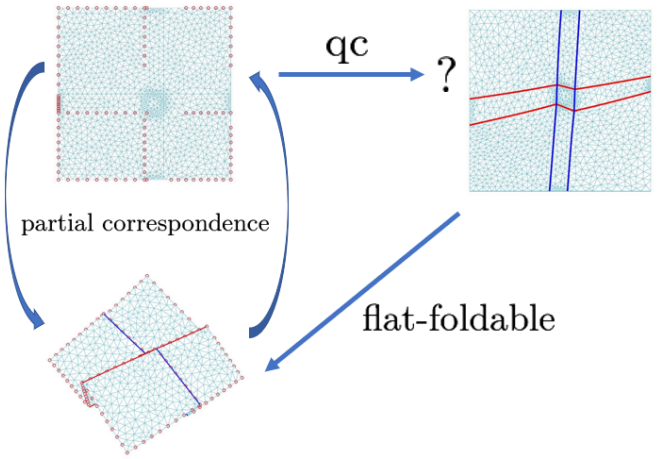


Figure 4. *The problem set-up of inferring the precise geometry of a partially observed flat-foldable surface. We assume partial correspondence between the observed fold and an initial domain with the same shape and topology of singular set configuration. The problem is how to find a quasi-conformal mapping that deforms the initial domain so that there is a generalized conformal map that maps the deformed domain while enforcing the correspondence. Here, the points in partial correspondence in shown in red circles.*

- We assume there is a domain of the same shape, with a singular set configuration of the same topology, and correspondence mapping (landmarks) between a subset of the domain and the same visible portion of the folded surface.

In practice the correspondence mapping is given on a subset of vertices from the boundary and the singular set. In Figure 4 we illustrate the problem setting.

As we signaled in the introduction, instead of finding the singular set directly, we will deform an existing one using the machinery developed so far. In the next section we shall formalize our approach to the recovery problem.

3.1. Our approach. To resolve the difficulty of incorporating Σ as a variable directly, we instead ask for a factorization of the flat-foldable folding homeomorphism f defined on the domain with desirable singular set Σ as follows:

$$\begin{array}{ccc} \Omega_{\Sigma_0} & \xrightarrow{\varphi} & \Omega_{\Sigma} \\ \downarrow{\scriptstyle g} & \swarrow{\scriptstyle f} & \\ S & & \end{array} \tag{9}$$

where Σ_0 is another singular set configuration of the same topology with Σ, g is a reference folding map that satisfies the given correspondence constraints, and φ is such that the diagram is commutative. Our approach is in a spirit similar to the classical Stoilow factorization [1]. Formally, the optimization problem is

$$\arg\min_{f,\varphi} \left(E(f,\varphi) := E_{\mathrm{GQC}}(u,v,\mu)\right), \tag{10}$$

where $f = (u, v), g = (\tilde{u}, \tilde{v})$ are such that $\tilde{u} = u \circ \varphi, \tilde{v} = v \circ \varphi$, $\mu = 0$ on $\Omega^+_{\varphi(\Sigma_0)}$, and $\mu = \infty$ on $\Omega^-_{\varphi(\Sigma_0)}$, and the minimization is subject to the constraints

$$f\big|_{\Omega_{\text{vis}}} : \Omega_{\text{vis}} \to S_{\text{vis}}.$$

Now the argument f, defined on Ω, ranges in the set of folding homeomorphisms, and $\varphi : \Omega \to \Omega$ ranges in the set of quasi-conformal homeomorphisms. Note that if φ gives the "correct" singular set configuration, then the above energy vanishes for f that is generalized conformal.

Note that there might be multiple solutions satisfying the same partial constraints, after all we only have partial geometric information about the folded surface. To this end we define the admissibility of folded surfaces as below.

Definition 3.1. *Let S_{vis} be a set of partial boundary and singular set data, and Ω_{Σ_0} be the domain with an initial singular set configuration. A folded surface S is called admissible if there is the following:*

1. *Topological equivalence: the singular set configuration of S is of the same topological type with the target surface.*
2. *Data correspondence: there is a subset $C \subset S$ such that there is an isometry from C to S_{vis}.*
3. *Cycle consistency: there exist maps g, φ, f such that f is flat foldable, and the diagram (9) commutes.*

3.2. The reinforcement iteration. An important point in the factorization (9) above is that if Σ_0 is close to Σ, then both the mappings g and φ should be close to being generalized conformal. Based on this observation, we now proceed to describe a fixed-point-like algorithm of finding some admissible folded surface S and its flat-foldable parametrization simultaneously. The iteration consists of two key steps, which find improved folding and unfolding maps given the current unfolded and folded surfaces in an alternating fashion. Intuitively, each unfolding step tries to find a better singular set configuration, while each folding step tries to conform with the given data.

Specifically, let n be the current iteration number, $\Omega_{\Sigma_n} = \Omega_{\varphi_n(\Sigma_0)}$. We first construct the folding map $g_n : \Omega_{\Sigma_{n-1}} \to S_n$ by solving the discrete system of (7) with Beltrami coefficients

$$\mu_1 = \begin{cases} 0 & \text{in } \Omega^+_{\Sigma_n}, \\ \infty & \text{in } \Omega^-_{\Sigma_n}, \end{cases} \tag{11}$$

which satisfies the constraints

$$g_n\big|_{\Omega_{\text{vis}}} : \Omega_{\text{vis}} \to S_{\text{vis}},$$

where $\Omega_{\text{vis}} \subset \Omega_{\Sigma_(n-1)}$ is the corresponding subset corresponding to the partial boundary and singular set data S_{vis}. This step mainly promotes data fidelity.

A remark is in order on the choice of μ_1. Note that because the constraints contain more than two points, we recall from Remark 2.15 that the solution will be a generalized quasi-conformal map unless the constraints are compatible with μ_1. In fact, as we shall see below, our goal is indeed to make them more compatible, so that during the iteration g_n will *become increasingly flat foldable*. We of course do not rule out the possibility that in the first iteration

one has already obtained some explicit data of the Beltrami coefficients and could have used them, but in the later iterations μ_1 is a consistent choice if g_n is becoming more flat foldable. In all examples shown in this paper we have adopted the above simple choice for μ_1.

In the next step we construct an unfolding map $h_n : S_n \to \Omega_{\Sigma_n}$, whose purpose is to get a deformed domain that contains an improved singular set configuration from the last folding map. To that end it is again obtained by solving the alternating Beltrami equation with

$$\mu_2 = \begin{cases} 0 & \text{in } S_n^+, \\ \infty & \text{in } S_n^- \end{cases} \tag{12}$$

enforcing the original shape constraints of Ω_{Σ_n}. This step minimize the generalized conformal distortion based on the fitted surface S_n. We note that both the S_{vis} and shape constraints are essential for convergence of the algorithm, in particular, implicitly decreasing the area distortion of the map.

The next map g_{n+1} is constructed based on the updated domain with its singular set configuration

$$\Sigma_n = h_n \circ g_n(\Sigma_{n-1}).$$

As $n \to \infty$, we want g_n to converge to a generalized conformal map $f : \Omega \to S$, and the composition

$$h_n \circ g_n := \psi_n$$

converges to id_Ω, meaning that

$$\psi_n \circ \cdots \circ \psi_2 \circ \psi_1 := \varphi_n$$

converges to a quasi-conformal map that transforms the initial singular set configuration to a desirable one. This is shown schematically in the following diagram:

$$\begin{array}{ccccccccccc} \Omega_{\Sigma_0} & \xrightarrow{\psi_1} & \Omega_{\Sigma_1} & \xrightarrow{\psi_2} & \Omega_{\Sigma_2} & \xrightarrow{\psi_3} & \cdots & \Omega_{\Sigma_{n-1}} & \xrightarrow{\psi_n} & \Omega_{\Sigma_n} & \cdots \\ \downarrow{\scriptstyle g_1} \nearrow{\scriptstyle h_1} & & \downarrow{\scriptstyle g_2} \nearrow{\scriptstyle h_2} & & \downarrow{\scriptstyle g_3} \nearrow{\scriptstyle h_3} & & & \downarrow{\scriptstyle g_n} \nearrow{\scriptstyle h_n} & & & \\ S_1 & & S_2 & & S_3 & & & S_n & & & \end{array} \tag{13}$$

In each step we keep enforcing the available data S_{vis} by the map g_n and, by h_n we keep enforcing the known boundary shape of Ω, hence the name reinforcement iteration. The overall algorithm is summarized as in Algorithm 3.1.

The basic steps are constructions of the maps g_n and h_n, whose implementations we now turn to.

3.3. Implementation details.

3.3.1. Construction of g_n. Given the updated domain $\Omega_{\Sigma_{\varphi_{n-1}}} = \varphi_{n-1}(\Omega_{\Sigma_0})$, we obtain the S_{vis}-enforced map g by solving an alternating Beltrami equations subject to the constraints

$$g\big|_{\Omega_{\text{vis}}} : \Omega_{\text{vis}} \to S_{\text{vis}}.$$

Algorithm 3.1 Reinforcement Iteration.

Inputs: Initialized domain Ω_{Σ_0}, partial data from folded surface S_{vis}.

Outputs: Optimal mapping φ^* such that $\varphi^*(\Omega_{\Sigma_0})$ has optimal singular set configuration.

$n = 1$, construct g_1, h_1; compute $\varphi_1 = u_1 \circ f_1$.
while not converged, **do**
  Provided φ_{n-1}, construct g_n, h_n using the partial data and domain shape constraints.
  Compute $\varphi_n = h_n \circ g_n \circ \varphi_{n-1}$.
  $n \leftarrow n + 1$.
end while

Recall that since we mainly aim for the partial boundary and singular data enforcement here, we simply set the Beltrami coefficients to be 0 or ∞ on the orientation-preserving or -reversing regions, respectively.

In terms of triangular meshes, suppose the domain triangular mesh is $\mathcal{D}_{n-1}$ with $\mathcal{D}^+_{n-1}$, $\mathcal{D}^-_{n-1}$ corresponding to Ω^+_{n-1} and Ω^-_{n-1}, respectively; S_{vis} is realized as certain constraint C_{vis}. Then the folded surfaced is obtained by

$$\mathcal{S}_{g_n} = \text{LSQC}(\mathcal{D}_{n-1}, \{\mu_T\}_{\mathcal{D}_{\varphi_{n-1}}}, C_{\text{vis}}),$$

where[2]

$$\mu_1 = \begin{cases} 0 & \text{if } T \in \mathcal{D}^+_{n-1}, \\ \infty & \text{if } T \in \mathcal{D}^-_{n-1}. \end{cases}$$

3.3.2. Construction of h_n. Given the folded surface constructed from the last step $S_n = g_n(\Omega_{n-1})$, recall that the unfolding map $h_n : S_n \to \Omega_{\Sigma_n}$ is found by solving the minimization problem

$$\arg \min_{h: S_n \to \Omega_{\Sigma_n}} E_{\text{GQC}}(h; \nu) \tag{14}$$

with ν as in (11), and subject to the shape constraints

$$h\big|_{\partial S_n} : \partial S_n \to \partial \Omega.$$

φ_n is then updated by $\varphi_n = h_n \circ g_n$. In terms of triangular meshes, suppose the folded surface mesh is $\mathcal{S}_n$ with $\mathcal{S}^+_n$, $\mathcal{S}^-_n$ corresponding to S^+_n and S^-_n, respectively; $\partial\Omega$ is realized as certain constraint $C_{\partial\Omega}$. Then the above minimization can be solved by

$$\mathcal{D}_{\Sigma_n} = \text{LSQC}(\mathcal{S}_n, \{\nu_T\}_{\mathcal{S}_n}, C_{\partial\Omega}),$$

where

$$\mu_2 = \begin{cases} 0 & \text{if } T \in \mathcal{S}^+_n, \\ \infty & \text{if } T \in \mathcal{S}^-_n. \end{cases}$$

[2]In practice we take a complex number with large enough modulus (e.g., $\mu = 10^5$) in place of ∞.

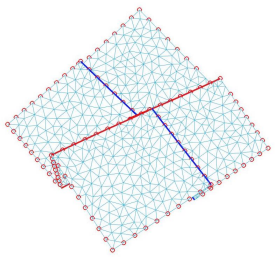
4. Further discussion and experimental results. We note that the desirable domain Ω^* is a fixed point of our iteration algorithm, which we write as $\Omega^* = \mathcal{F}(\Omega^*)$, where $\mathcal{F}$ is the iteration map in operator form. Note that $\mathcal{F}$ depends on its argument Ω in a very nonlinear way because of the intermediate folded surface S_n we introduced in each iteration, whose computation requires the cotangent matrix associated with Ω_{Σ_n}. But approximately, in each iteration the folding and unfolding operations are inverse to each other and, therefore, $\mathcal{F}$ is close to the identity. The convergence of fixed-point iterations is well studied in the literature; see [6] and references therein. For example, the convergence will be implied by the α-averaged property of $\mathcal{F}$. As we can notice in Figure 5, as well as in many other experiments, the distortion of many of the interior mesh triangles can barely be noticed in the later phase of the iteration, while the meshes also remain well conditioned. As for other fixed-point iterations, it is reasonable to expect that the iteration map under good conditioning of the mesh triangles, with enough constraints, and a good initial guess, to have convergence.

We implemented the described reinforcement iteration algorithm and demonstrate it for a doubly folded surface, as illustrated in Figure 5. The folded surface and its unfolded counterpart, as shown in Figures 5(a) and 5(b), are generated according to a real folded paper and its unfolded counterpart. In Figure 5(c) it is our initialized domain Ω_{Σ_0}. In 5(a) and 5(c), the red circles mark the corresponding constraint points to the visible partial singular set and boundary data. Our algorithm works similarly well with other examples as well. This shows the robustness of our algorithm.

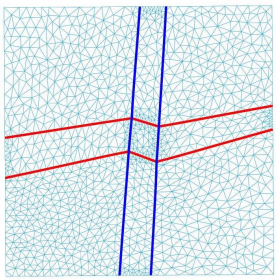
In the next three rows of Figure 5 we show the iteration results at the first iteration, 50th iteration, and 200th iteration. We can observe the curly folding lines in the first iteration in Figures 5(e) and 5(f). This is due to the incomplete data and the incompatible initialized domain. In the subsequent iterations we saw significant improvement over the rigidity of the folding. In practice we also found that if in the later phase of the iteration, we explicitly regularize the singular lines by fitting the vertices into a Euclidean geodesic in a least squares sense, and then restart the iteration, the convergence will have some minor speed-up, in particular, for the heavily multiply folded cases.

Observe also that in the limit, as in Figure 5(l), the singular set configuration is not exactly the same as that of the true unfolded surface. This can be explained by the existence of multiple admissible solutions to this problem. For example, another admissible solution may be obtained by some different initialization. This is of course expected.

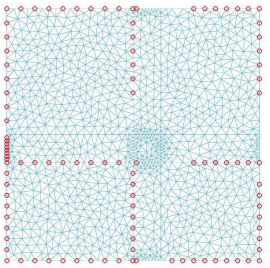
In Figures 6 and 7 we illustrate the effect of reinforcement iteration algorithm applied to a once-folded surface and a cusped surface. The straightening effect can be easily seen from the comparison between the initial folding map and the final folding map. In Figure 8, we plot a log-log diagram for the scale insensitive version of the energy $E(g_k, \varphi_k)$ for the examples in Figures 5 and 7. The plot here uses only the iteration algorithm and no other regularization. We can observe that the convergence rate approaches $O(1/N)$ in the midstage of the iteration. That the energy decreases slightly slower in the later phase can be explained by our observation from the iterations that only a few points are adjusted while the singular set configuration is still away from flat foldability. These adjusted points are mainly near the cusp points. This fact can be observed from Figures 5(f) and 5(i). The convergence rate varies in the different phases of the iteration, illustrating the nonlinear nature of the iteration.



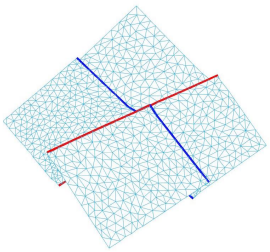
(a) Folded surface with occlusion.



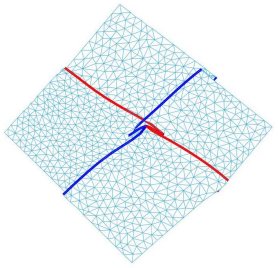
(b) True unfolded surface.



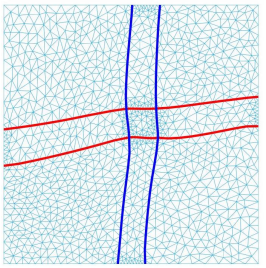
(c) Initialized domain with partial data.



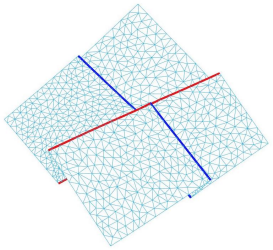
(d) Iter = 1: Frontside of registered fold.



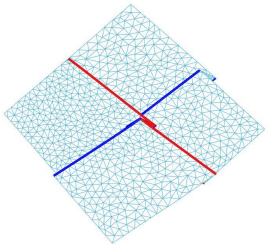
(e) Iter = 1: Backside of registered fold.



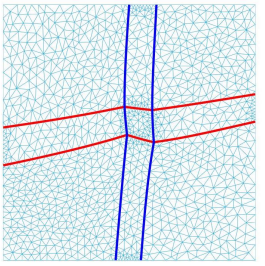
(f) Iter = 1: Unfolded surface.



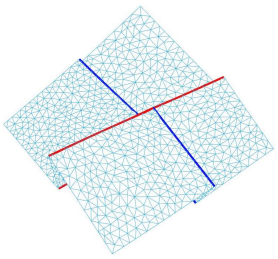
(g) Iter = 50: Frontside of registered fold.



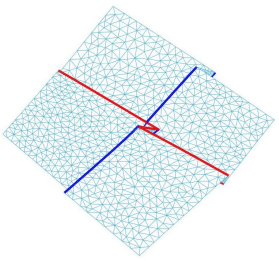
(h) Iter = 50: Backside of registered fold.



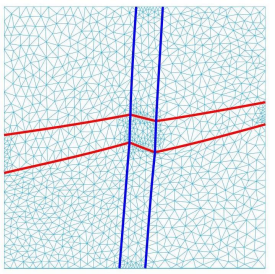
(i) Iter = 50: Unfolded surface.



(j) Iter = 200: Frontside of registered fold.

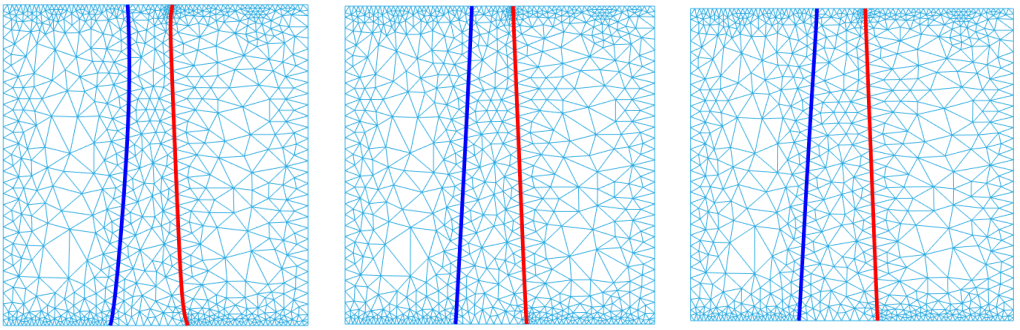


(k) Iter = 200: Backside of registered fold.

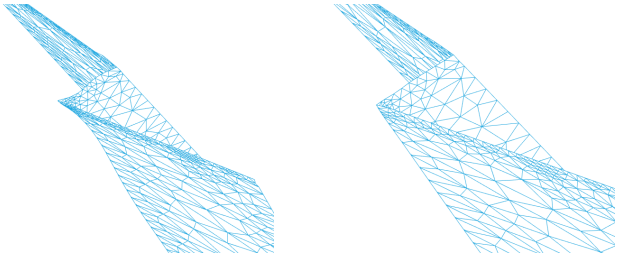


(l) Iter = 200: Unfolded surface.

Figure 5. *Iteration results for the doubly folded surface: note that the folding lines gradually straighten out.*



(a) Iter = 1: unfolded domain. (b) Iter = 50: unfolded domain. (c) Iter = 200: unfolded domain.



(d) Iter = 1: initially registered 1-fold. (e) Iter = 200: finally registered 1-fold.

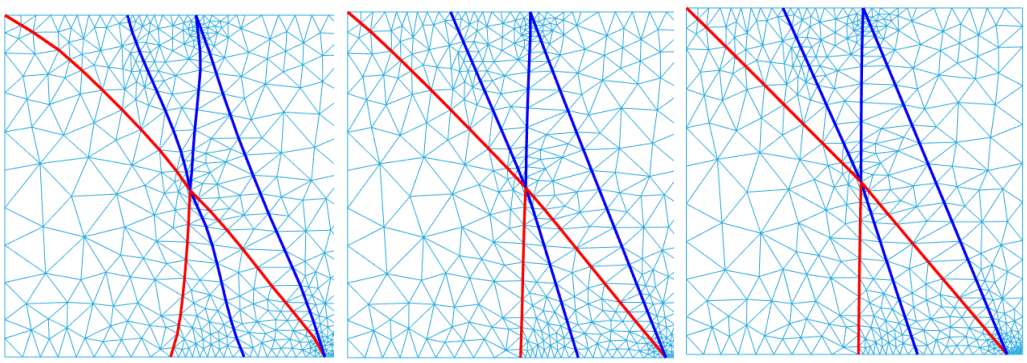
Figure 6. *Iteration results for a once-folded surface: note the curved boundary in occlusion from the initial map is gradually straightened out and the folded domain becomes wider.*

5. Applications.

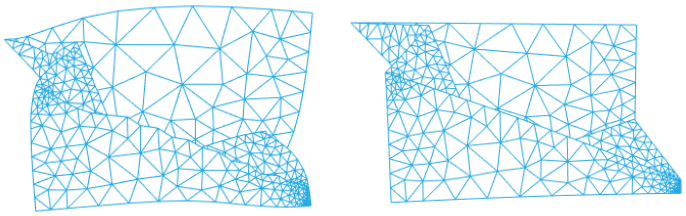
5.1. Generating and editing generalized Miura-ori. The Miura-ori refers to a special type of origami tessellation of the plane, which can be used to design flat-foldable materials aiming at achieving designed curvature properties [10]. Previous approaches are based on analytic construction or constrained optimization, using the Kowasaki condition. Here we explore another possibility of creating such origami models. Namely, we create more Miura-ori type domains and realize them via solving alternating Beltrami equations.

For simplicity, we consider the Miura-ori pattern in Figure 9. The yellow color on a triangle T refers to the prescription of $\mu(T) = \infty$, and purple ones $\mu(T) = 0$. After solving the alternating Beltrami equation in 2 dimensions by pinning two vertices, we obtain the classical Miura-ori strip, which is the flat-folded state of the surface. Suitable z-coordinates are added for visualization in 3 dimensions.

To generate more Miura-ori type domains, ideally we can simply apply a conformal map on the domain. Notice that, in the continuous case, the domain obtained by compositing a flat-foldable configuration with a conformal map remains flat foldable (satisfying the alternating Beltrami equation with the same coefficients as before), because of the angle preserving property. Such a composition can create triangles at different scales.



(a) Iter = 1: unfolded domain. (b) Iter = 50: unfolded domain. (c) Iter = 200: unfolded domain.



(d) Iter = 1: initially registered cusp. (e) Iter = 200: finally registered cusp.

Figure 7. *Iteration results for a cusped surface: note that the curved boundary and folding lines in occlusion from the initial map are gradually straightened out.*

However, because of the discreteness, the angles are preserved only if the map is a uniform scaling plus rigid motion. Indeed, this follows from our assertion on rank of the system matrix. Fortunately, applying a conformal map usually only yields a small and structured perturbation to the Beltrami equation, and the new Miura-ori domains can still be created via several iterations of the foldings and unfoldings, in light of the reinforcement iteration we proposed. For example, a new Miura-ori pattern in Figure 10 is created via this method, with the choice of (in this case we just made any convenient choice)

$$\Phi(z) = 10 + 0.1z + 0.4z^2.$$

Different from the approach of Dudte et al. [10], the surface we obtain is flat foldable by design. Given the rich family of conformal maps, it will be particularly interesting to study the new family of Miura-ori patterns with the aid of our algorithm. The study of different patterns' curvature approximation capacities is also an exciting future direction. We envisage a "conformal geometric processing" approach to the modeling of Miura-ori. Under such an approach researchers can efficiently design the pattern with a simple set of computer-aided design tools. Mathematical understanding of this problem will definitely benefit such a "bottom-up" approach to material design with flat-foldable structures.

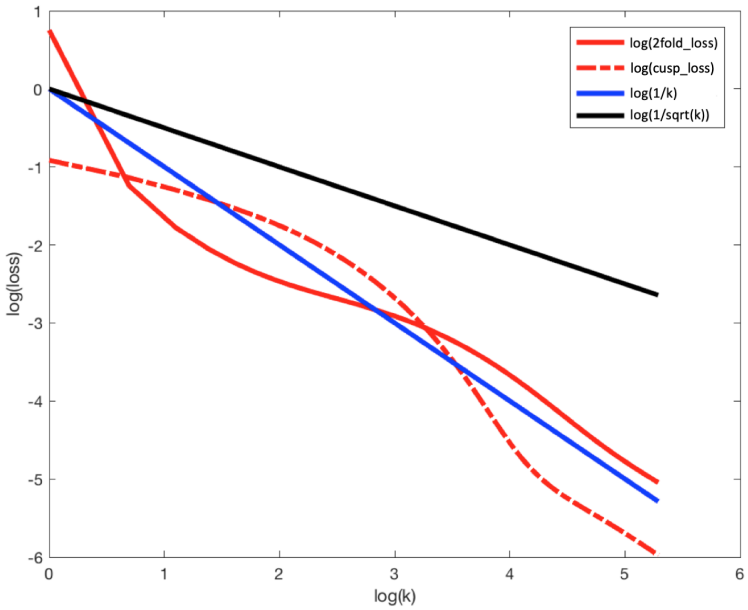
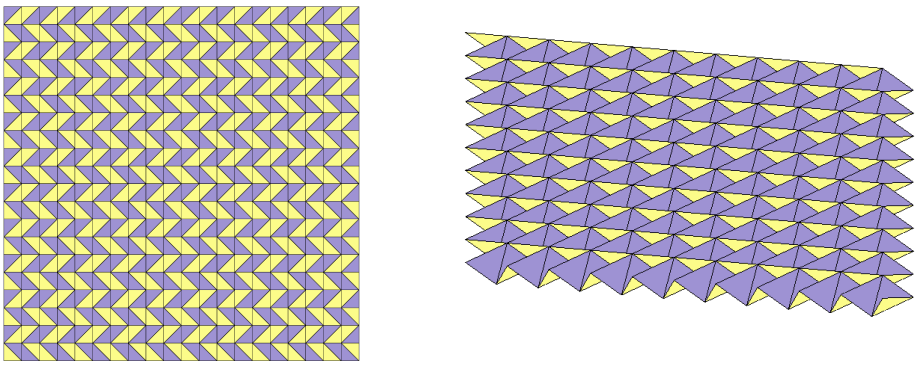
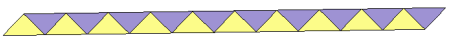


Figure 8. *Convergence plot: the loss here is defined as* $\sum_{T\in\Omega_+}|\mu_{g_n}|^2+\sum_{T\in\Omega_-}1/|\mu_{g_n}|^2$ *for scale invariant comparison, where* μ_{g_n} *denotes the Beltrami coefficient associated with the mapping* g_n*, which should be close to* 0 *(or* ∞*) on* Ω^+ *(or, respectively, on* Ω^-*).*

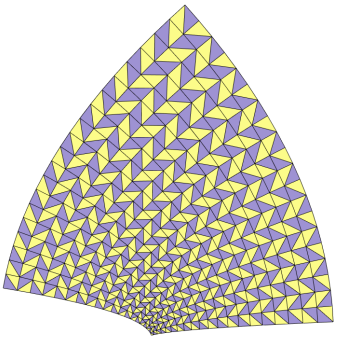


(a) Classical Miura-ori pattern. (b) Realization of the Miura-ori on the left.

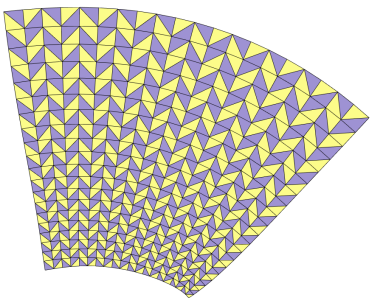


(c) Classical Miura-ori strip.

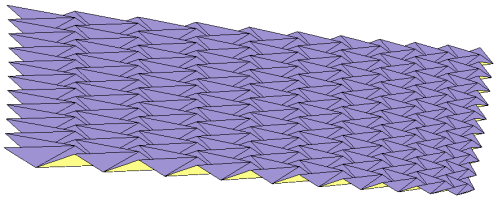
Figure 9. *Classical Miura-ori and its realization in* 3 *dimensions.*



(a) Naive composition with Φ.



(b) A new Miura-ori pattern after folding-unfolding iterations (maximal distortion $= 3 \times 10^{-4}$ compared to the folded state (c)).



(c) Realization of the Miura-ori in (b).

(d) New Miura-ori strips: note that there is a stack of strips of varing sizes.

Figure 10. *A new Miura-ori pattern by composition with* Φ *and its realization in* 3 *dimensions. Here, maximal distortion is defined by* $\max\{\max\{|\mu_T|\}_{T\in\Omega^+}, \max\{1/|\mu_T|\}_{T\in\Omega^-}\}$.

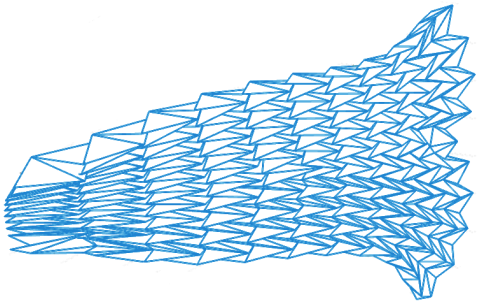
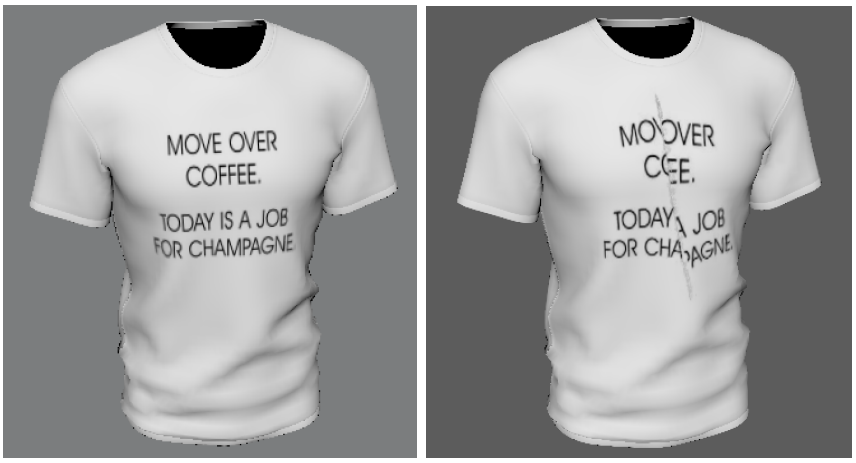


Figure 11. *A rigid deformation of the Miura-ori in Figure* 10.

For a preliminary example, we can simulate and study the deformation of the Miura-ori in 3 dimensions with our solutions. Starting from the flat-folded state of the surface, one can apply the classical geometric editing methods such as as-rigid-as-possible [21]. An example of such a deformation with user-defined position constraints is shown in Figure 11.



(a) Before fold texturing. (b) After fold texturing.

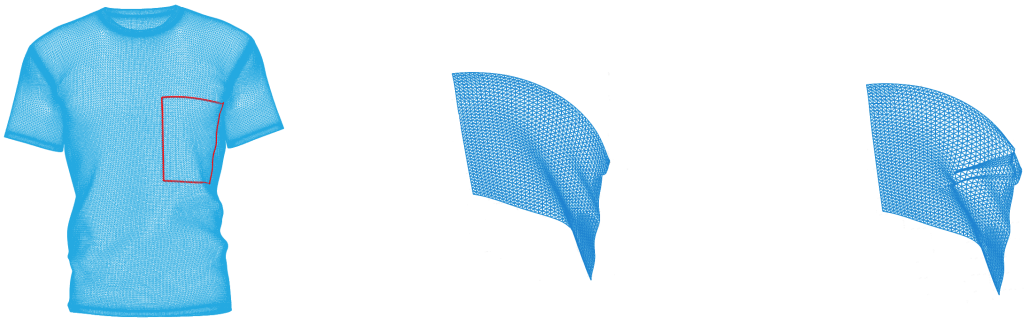
Figure 12. *Folding effect texturing on a 3D model. Note that the mesh model is not deformed.*

5.2. Almost rigid folding with application to fold-texture generation, fold sculpting, and fold in-painting. As one of the immediate applications, we can consider a folding transformation on the texture space to create synthetic foldlike textures, prior to applying the texture map. This can be cheap to do if high quality physical simulation and rendering is not available. In Figure 12 we explore such a possibility of user-designed foldlike texture generation.

One of the fundamental steps in texturing a 3D surface is to find the parametrization (or the texture map) $f : S \to \Omega \in \mathbb{R}^2$. In particular, an ultraviolet (UV) map is one of the major types of parametrization techniques in various software packages, which works well if the 3D model is created from polygon meshes. The above technique can be very useful in the interactive user design, where the user directly operates on the target mesh, and the input is transformed to the texture domain via the UV map, to create desirable foldlike texture on the target mesh. It is also possible to incorporate a proper shading effect on the transformed texture directly, making the texture look more realistic. We have implemented such a foldlike texturing method using a 3D T-shirt model, shown in Figure 12(b). Note that the mesh is not deformed at all.

We can also apply the folding technique directly to the 3D meshes, as an application we would like to call *fold sculpting*. To illustrate this, we select a patch from the T-shirt model, as shown in Figure 13(a). We applied the folding operation to a suitable parametrization of the patch, which can obtained easily via, for example, projection or a least squares conformal parametrization [15], and then glue it back to the T-shirt model. Note that our algorithm produces sharp edges. This can be mitigated by some standard smoothing operation in various mesh editing software. Figures 14 and 15 show the results after appropriate smoothing, where we used the software Maya[3] to do the smoothing and rendering tasks. Note that such folding

[3]A software of Autodesk Inc. See https://www.autodesk.com.hk/products/maya/overview. The results are generated under the student license obtained by the first author.



(a) The T-shirt model. (b) The original patch. (c) The deformed patch.

Figure 13. *Patchwise fold sculpting: the region inside the red contour is the patch selected; appropriate alternating Beltrami equation is then solved in the patch domain to obtain the desired folding effect.*

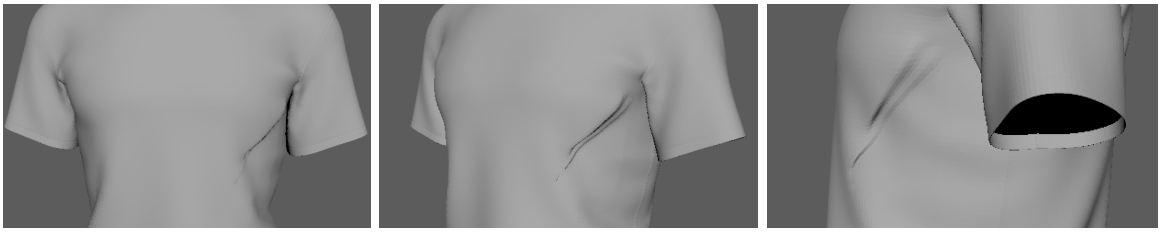


Figure 14. *Results of fold sculpting on the T-shirt model after appropriate smoothing: two short folds are sculpted on the right.*

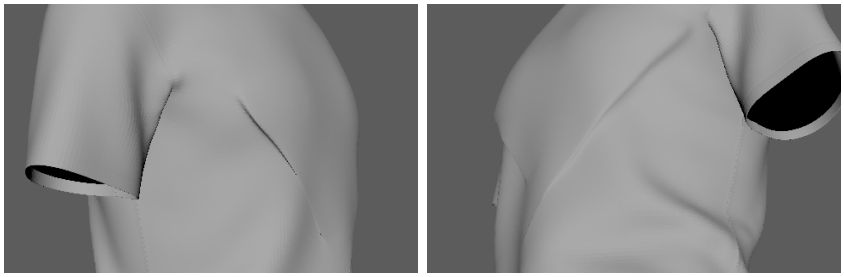


Figure 15. *Results of fold sculpting on the T-shirt model: two long folds are sculpted on the left and right.*

is not easily obtained by pure handcraft, since one part of the cloth actually folds over and covers some other part of the cloth.

The technique can also be applied after the acquisition of a folded surface using laser scans, where the folded part introduces self-occlusions and the folding is usually diminished or destroyed after applying the watertight operation. To preserve the folding details from the scans directly, we can mark the folding part that we want to preserve in the raw acquisition. By taking a patch like we did before and mapping it into the plane, we can solve a proper alternating Beltrami equation to obtain the desired folding effect. The folded patch can then

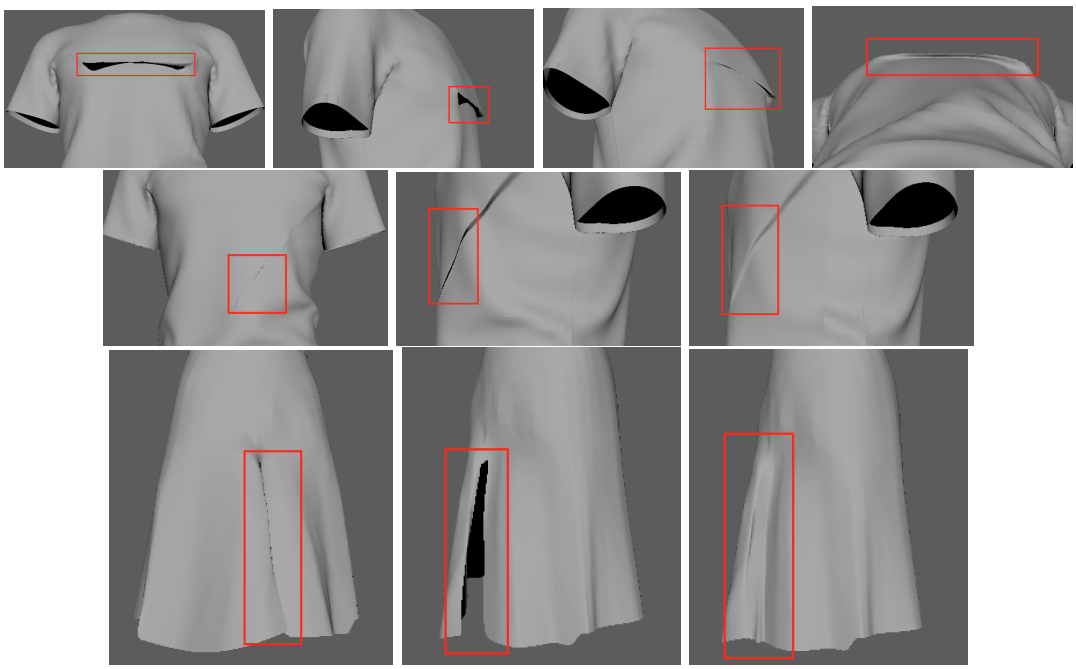


Figure 16. *Results of fold in-painting. For each row, the two on the left are the surface with holes from acquisition due to self occlusion. The results of in-painting are on the right. The corresponding holed and in-painted regions are highlighted inside the red boxes.*

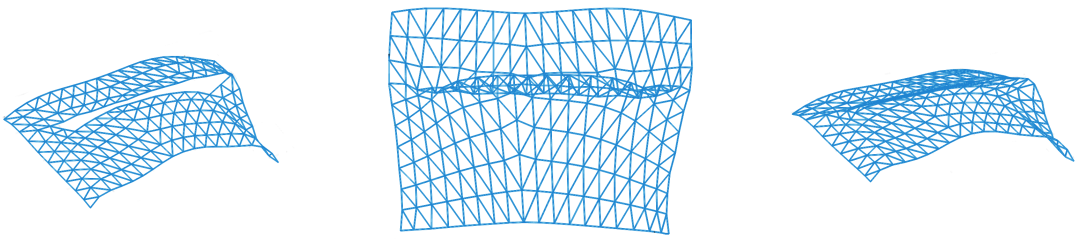


Figure 17. *Patchwise fold in-painting: from the raw acquisition of a self-occluded surface, we map the occluded, holed surface to the plane and fill the hole; we then apply a suitable folding operation to reproduce the fold that was not captured during acquisition.*

be mapped back to the raw acquisition. To illustrate this, we have done a synthetic experiment using the above approach. We begin with an incomplete acquisition of a shirt model and a dress model, such as the one on the left in Figure 16. As demonstrated in Figure 17, the patch with holes is the first map to the plane and subsequently filled. A suitable folding operation is then applied to the patch to produce a plausible fold geometry given the acquisition data. The reconstruction is shown in Figure 16 on the right.

5.3. Self-occlusion reasoning of flat-foldable surfaces and its application to restoration of folded images. Given a single perspective of a folded surface, for example, shown in

(a) mountain (b) The Sower (c) building

Figure 18. 1-*fold,* 2-*fold, and cusp-fold examples.*

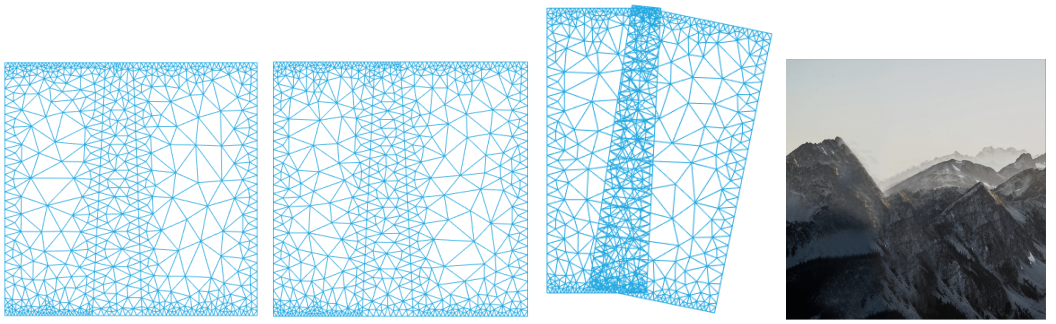
(a) 1-fold

(b) 2-fold (c) cusp-fold

Figure 19. *Occluded regions of various folded paper examples.*

Figure 18, we can use the proposed reinforcement iteration to unfold the surface, thus enabling us to identify the self-occluded region in the unfolded domains, shown in Figure 19.

Given a folded image, the task of restoring the image involves unfolding the image and in-painting the missing parts beneath the folded region. The performance of the final restoration results obviously depends on the realization of the texture synthesis. However, it is worth noting that the unfolding result may also drastically affect the in-painting result since many



(a) Ground truth. (b) Unfolded mesh. (c) Parametrized. (d) Restored result.

Figure 20. *Unfolding and restoring result of a* 1*-folded image.*

in-painting methods, such as the diffusion-based [4, 23] and the exemplar-based [7, 2, 3], assume the full knowledge of the computational domain (i.e., the image domain). To alleviate the difficulty arose from the incomplete knowledge of the image domain, we can employ the proposed unfolding algorithm to retrieve the geometric information of the folded subdomain with the given partial geometric information. Once we restore the intrinsic image domain consistent with the partial geometric information, well-developed in-painting techniques can then be employed correctly and provide satisfactory in-painting results.

Figure 20 shows the result for the 1-fold example. The unfolding is trivial in a sense, but here we want to use it to illustrate the typical procedure of a folded-image restoration. The same procedure applies to all kinds of folds alike. In any case, we assume to have the partial boundary and singular set data (i.e., the folding edges and the boundaries) available from the folded images. Our goal is to recover the folded image Figure 18(a) by using the proposed unfolding technique and some well-established in-painting algorithms. At the beginning of our algorithm, an initialization Σ_0 is constructed by simply using the partial boundary and singular set data. By the proposed Algorithm 1, we can successfully reconstruct the folding map based on the reinforcement iterations, which are shown in Figure 20(b). With the folding map, we can obtain the occluded region and carry out the in-painting procedure. Notice that the unfolded mesh in Figure 20(b) highly resembles the ground truth (see Figure 20(a)). With such an unfolded domain, we can acurately approximate the masked region and apply the patch-matching-based in-painting algorithm to recover the image. The corresponding parametrized folding surfaces are show in Figure 20(c). The image can then be mapped from the folded surface back to the domain complementary to the occluded region. Here, as is common, the case, due to the size of masked region generated from the fold of the image, we choose the patch-matching-based algorithm for image in-painting. In particular, we employ the algorithm[4] proposed by Daisy, Tschumperlé, and Lézoray. [7] in this example. The result is shown in Figure 20(d). For comparison, the original image together with the overlapping mask (drawn as a half-transparent domain) is shown in Figure 19(a).

[4]The algorithm is available as a plugin for the open source GIMP2 software. The software is available at https://www.gimp.org/.

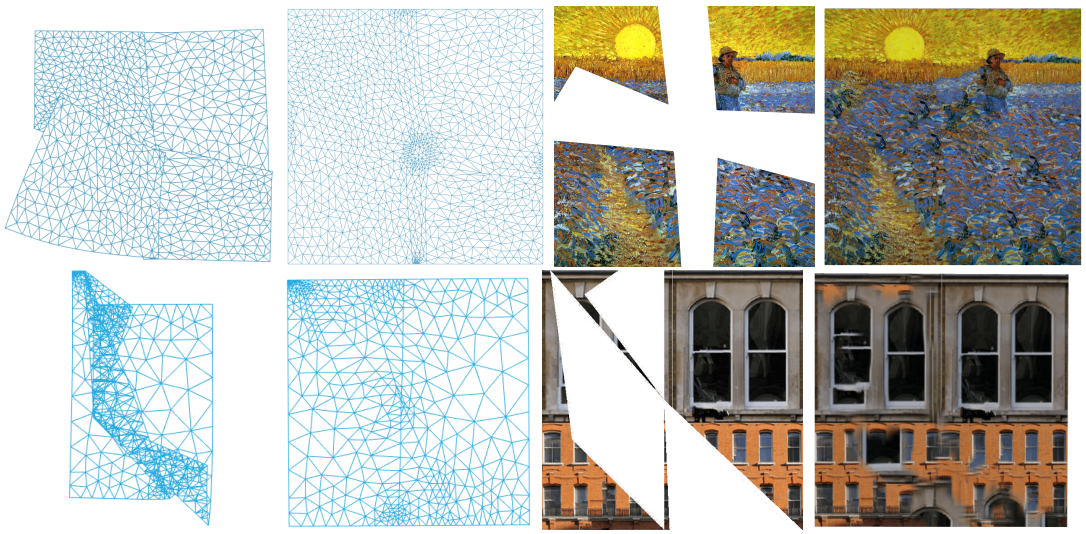


Figure 21. *Unfolding and in-painting results for the 2-fold (first row) and cusp-fold (second row) examples. Leftmost column shows the folded meshes representing the domain of the images. The unfolded results are shown in the middle-left column and the recovered occluded domains are shown in the middle-right columns. The overall recovery results are shown in the rightmost column.*

To illustrate the adaptability of our proposed algorithm in some more complicated folded surface, we now consider correspondingly, 2-, 3-folded examples, as shown in Figure 18. There is a 2-folded painting "The Sower" by Vincent van Gogh, and a cusp-folded "building" image.

Similarly, to approximate the original images with the given folded data, we first have to unfold these surfaces using the proposed algorithm. Unlike the trivial 1-fold example illustrated above (which may be simply unfolded even without the use of Algorithm 1), the folding order also takes part in this unfolding problem since different orders of folding produce different folded images. When the folding number is large, obtaining the ordering of the folds from the given data is difficult. However, using Algorithm 1, this folding order can be obtained implicitly. In other words, the unfolding procedure using Algorithm 1 is fully automatic. Figure 21 shows the unfolding and the corresponding in-painting results. The leftmost column shows the folded meshes corresponding to some unidentified rectangular meshes. With only partial boundary conditions and singular set data, unfolding these surfaces is highly ill-posed. But using our proposed algorithm, we successfully obtained the unfolding surfaces (the middle-left column). Notice that by regularizing the generalized Beltrami coefficient, Algorithm 1 converges to unfolded regular meshes, where unnatural curvy edges are not presented. With these unfolded meshes, we can recover the occluded regions due to the foldings (see the middle-right column) and therefore in-painting algorithms can be employed as usual. The overall recovered images are shown in the rightmost column of Figure 21.

6. Discussion and conclusion. We have proposed a novel way of studying and modeling the folding phenomena of surfaces using alternating Beltrami equations. The numerical scheme is proposed to discretize and solve the equation, by taking into account the coupled nature of the two coordinate functions of the solution. The resulting method works for as few as fixing

two point as constraints, and has a nice geometric interpretation. More importantly, it allows us to formulate and solve the inverse problem of inferring and parametrizing flat-foldable surfaces with observed partial data. We have proposed to use the reinforcement iteration algorithm in order to solve the associated optimization problem, which has shown empirical convergence over various examples. Various applications are given, including fold sculpting, foldlike texture generation, generating and editing generalized Miura-ori patterns, as well as self-occlusion reasoning. Many more possible applications shall be explored in the future. At the same time, the understanding of nonrigid folding is still largely incomplete and many more interesting examples and applications are waiting to be discovered.

Acknowledgments. The first author would like to thank Mr. Leung Liu Yusan and Dr. Emil Saucan for some useful help and discussions in the early stage of this work. The examples' meshes are generated by the software Triangle [20].

REFERENCES

[1] K. Astala, T. Iwaniec, and G. Martin, *Elliptic Partial Differential Equations and Quasiconformal Mappings in the Plane (PMS-48)*, Princeton University Press, Princeton, NJ, 2008.

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, *Patchmatch: A randomized correspondence algorithm for structural image editing*, ACM Trans. Graph., 28 (2009), 24.

[3] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, *The generalized patchmatch correspondence algorithm*, in European Conference on Computer Vision, Springer, Berlin, 2010, pp. 29–43.

[4] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, *Image inpainting*, in Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, ACM, New York, 2000, pp. 417–424.

[5] P. T. Choi, K. C. Lam, and L. M. Lui, *FLASH: Fast landmark aligned spherical harmonic parameterization for genus-0 closed brain surfaces*, SIAM J. Imaging Sci., 8 (2015), pp. 67–94.

[6] P. L. Combettes, *Solving monotone inclusions via compositions of nonexpansive averaged operators*, Optimization, 53 (2004), pp. 475–504.

[7] M. Daisy, D. Tschumperlé, and O. Lézoray, *A fast spatial patch blending algorithm for artefact reduction in pattern-based image inpainting*, in SIGGRAPH Asia 2013 Technical Briefs, ACM, New York, 2013, 8.

[8] E. D. Demaine and T. Tachi, *Origamizer: A practical algorithm for folding any polyhedron*, in 33rd International Symposium on Computational Geometry, LIPIcs-Leibniz Int. Proc. Inform. 77, Leibniz-Zentrum fuer Informatik, Schloss Dagstuhl, 201, 34.

[9] M. Desbrun, M. Meyer, and P. Alliez, *Intrinsic parameterizations of surface meshes*, in Comput. Graph. Forum 21, (2002), pp. 209–218.

[10] L. H. Dudte, E. Vouga, T. Tachi, and L. Mahadevan, *Programming curvature using origami tessellations*, Nature Mater., 15 (2016), pp. 583–588.

[11] X. Gu, Y. Wang, T. F. Chan, P. M. Thompson, and S.-T. Yau, *Genus zero surface conformal mapping and its application to brain surface mapping*, IEEE Trans. Med. Imag., 23 (2004), pp. 949–958.

[12] V. Gutlyanskii, V. Ryazanov, U. Srebro, and E. Yakubov, *Alternating Beltrami equation*, The Beltrami Equation, Springer, New York, 2012, pp. 183–208.

[13] M. Kilian, S. Flöry, Z. Chen, N. J. Mitra, A. Sheffer, and H. Pottmann, *Curved folding*, ACM Trans. Graph., 27 (2008), 75.

[14] T.-H. Kwok, C. C. Wang, D. Deng, Y. Zhang, and Y. Chen, *Four-dimensional printing for freeform surfaces: Design optimization of origami and kirigami structures*, J. Mech. Des., 137 (2015), 111413.

[15] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, *Least squares conformal maps for automatic texture atlas generation*, ACM Trans. Graph., 21 (2002), pp. 362–371.

[16] M. Li, *FoldSketch: Enriching Garments with Physically Reproducible Folds*, PhD thesis, University of British Columbia, Vancouver, Canada, 2018.

[17] L. M. Lui, K. C. Lam, S.-T. Yau, and X. Gu, *Teichmuller mapping (t-map) and its applications to landmark matching registration*, SIAM J. Imaging Sci., 7 (2014), pp. 391–426.
[18] L. M. Lui, T. W. Wong, W. Zeng, X. Gu, P. M. Thompson, T. F. Chan, and S.-T. Yau, *Optimization of surface registrations using Beltrami holomorphic flow*, J. Sci. Comput., 50 (2012), pp. 557–585.
[19] U. Pinkall and K. Polthier, *Computing discrete minimal surfaces and their conjugates*, Exp. Math., 2 (1993), pp. 15–36.
[20] J. R. Shewchuk, *Triangle: Engineering a 2d quality mesh generator and delaunay triangulator*, in Applied Computational Geometry Towards Geometric Engineering, Springer, Berlin, 1996, pp. 203–222.
[21] O. Sorkine and M. Alexa, *As-rigid-as-possible surface modeling*, in Eurographics Symposium on Geometry Processing, Vol. 4, Eurographics Association, Aire-la-Ville, Switzerland, 2007, pp. 109–116.
[22] U. Srebro and E. Yakubov, *Branched folded maps and alternating Beltrami equations*, J. Anal. Math., 70 (1996), pp. 65–90.
[23] D. Tschumperle and R. Deriche, *Vector-valued image regularization with PDEs: A common framework for different applications*, IEEE Trans. Pattern Anal. Mach. Intell., 27 (2005), pp. 506–517.
[24] R. Zayer, C. Rossl, and H.-P. Seidel, *Discrete tensorial quasi-harmonic maps*, in Shape Modeling and Applications, 2005 International Conference, IEEE Computer Society, Los Alamitos, CA, 2005, pp. 276–285.
[25] L. Zhu, T. Igarashi, and J. Mitani, *Soft folding*, Comput. Graph. Forum, 32 (2013), pp. 167–176.