

# Weaving Geodesic Foliations

JOSH VEKHTER, The University of Texas at Austin  
JIACHENG ZHUO, The University of Texas at Austin  
LUISA F GIL FANDINO, The University of Texas at Austin  
QIXING HUANG, The University of Texas at Austin  
ETIENNE VOUGA, The University of Texas at Austin

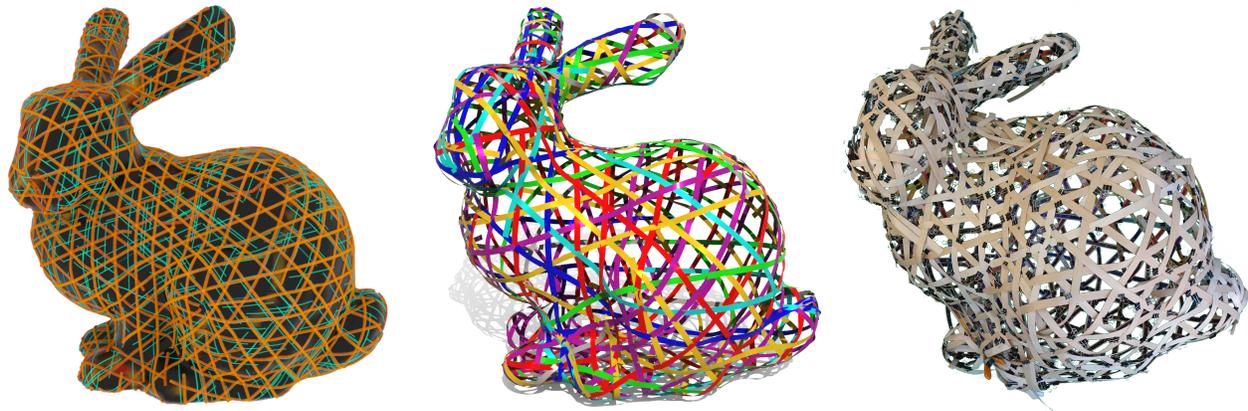


Fig. 1. We present new algorithms for computing geodesic foliations on discrete surfaces. A geodesic foliation on a six-fold branched cover of the Stanford bunny (left) describes the ribbon layout of a triaxial weave (center), which we fabricate out of birch veneer (right).

We study discrete geodesic foliations of surfaces—foliations whose leaves are all approximately geodesic curves—and develop several new variational algorithms for computing such foliations. Our key insight is a relaxation of vector field integrability in the discrete setting, which allows us to optimize for curl-free unit vector fields that remain well-defined near singularities and robustly recover a scalar function whose gradient is well aligned to these fields. We then connect the physics governing surfaces woven out of thin ribbons to the geometry of geodesic foliations, and present a design and fabrication pipeline for approximating surfaces of arbitrary geometry and topology by triaxially-woven structures, where the ribbon layout is determined by a geodesic foliation on a sixfold branched cover of the input surface. We validate the effectiveness of our pipeline on a variety of simulated and fabricated woven designs, including an example for readers to try at home.

CCS Concepts: • **Computing methodologies** → **Mesh geometry models**; Continuous simulation; • **Mathematics of computing** → *Geometric topology*.

Authors' addresses: Josh Vekhter, The University of Texas at Austin, 2317 Speedway, Austin, TX, 78712; Jiacheng Zhuo, The University of Texas at Austin; Luisa F Gil Fandino, The University of Texas at Austin; Qixing Huang, The University of Texas at Austin; Etienne Vouga, The University of Texas at Austin.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
0730-0301/2019/7-ART34 \$15.00  
<https://doi.org/10.1145/3306346.3323043>

Additional Key Words and Phrases: Discrete Differential Geometry. Vector Field Design. Digital Fabrication. Texture Synthesis.

## ACM Reference Format:

Josh Vekhter, Jiacheng Zhuo, Luisa F Gil Fandino, Qixing Huang, and Etienne Vouga. 2019. Weaving Geodesic Foliations. *ACM Trans. Graph.* 38, 4, Article 34 (July 2019), 22 pages. <https://doi.org/10.1145/3306346.3323043>

## 1 INTRODUCTION

Birds in the family *Ploceidae* have been weaving their nests since the Miocene; humans likewise have been using weaving to construct baskets, mats, and other two and three-dimensional aspects of the built environment for at least 10,000 years. More recently, composites woven out of non-traditional materials like carbon fiber and shape memory alloy have been gaining popularity in a number of domains, where improved material properties, as well as cost-effectiveness and the increasing capability of looms to weave in three dimensions, make it a particularly attractive technique for fabrication. Figure 2 shows some modern applications of weaving to architecture, art, chemical engineering, and medicine.

Away from singularities, fibers of the weave can be grouped into nearly-parallel families—two families at right angles in the case of simple weaves (plain, twill) and three interleaved at sixty degree angles for *triaxial* weaves. In their simplest form, all of these weave patterns are planar; curvature can be introduced into the woven structure by varying the spacing between consecutive members of the ribbon families, varying the angles at which two families cross, or inserting singularities or *dislocations* into the weave pattern. Such

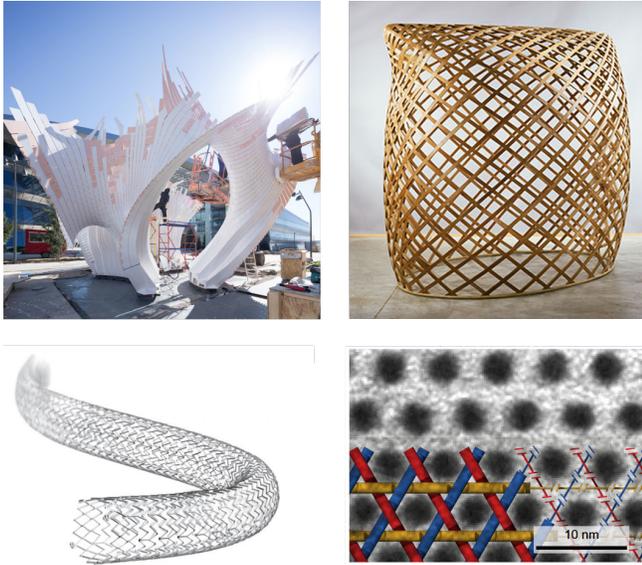


Fig. 2. Inspirations: thanks to their simplicity, beauty, and deployability, triaxial weaves have seen application in diverse areas such as (clockwise from top-left): architecture [Fornes 2017], art [Puryear 1998], nanochemistry [Lewandowska et al. 2017], and medicine [Boston Scientific 2015].

dislocations are desirable because they concentrate curvature, allowing for more symmetric and stable weaving patterns of complex surfaces that are simpler to fabricate and require less material.

The key principle relating the physics of a weave to its geometry is the following: ribbons are thinner than they are wide, and so bend much more readily out-of-plane (about their width axis) than in-plane (about their thickness axis). Attempting to force high in-plane curvature causes ribbons to buckle and twist instead of lying flush to a woven structure. Therefore the weave must be designed so that ribbons *lie along geodesics* of the target surface. We leverage this observation to cast the problem of laying out families of ribbons weaving an arbitrary curved surface as the geometry problem of covering the surface by one-parameter families of approximately-geodesic curves. Such *geodesic foliations* are interesting mathematical objects in their own right with a long history of study [Poincare 1905], and appear in the modeling of a wide array of phenomenon in physics, ranging from the orbital dynamics of objects in space [Klingenberg 1978] to the structural stability of thin shells under tension [Vandeparre et al. 2011] to the micro-magnetics of controlling LCD panels and other settings that exhibit “domain-wall” effects [Kohn 2006] [Machon et al. 2019].

**Contributions.** We systematically study the geometry processing problem of designing discrete geodesic foliations on surface meshes. Our contributions include:

- a vector field design algorithm, based on a discretization and relaxation of the geodesic equation, for computing geodesic vector fields on triangle meshes. This algorithm is global and handles nontrivial topology and field singularities;

- a novel method for global vector field integration, specifically designed to yield a global periodic function whose isolines align with the input vector field. Applying this second algorithm to fields computed using our first yields geodesic foliations;
- a design and fabrication pipeline for triaxial weaves approximate arbitrary target geometries, based on computing and integrating geodesic foliations on six-fold branched covers of the input surface.

## 2 RELATED WORK

**Vector Field Design.** The problem of optimizing vector fields on surfaces, for various purposes, is well-trodden ground in computer graphics. The survey by Vaxman et al [2016] provides a thorough overview. Methods vary in the objectives they optimize (smoothness, intrinsic and/or extrinsic feature alignment, etc) and in whether number and location of singularities are prescribed [Crane et al. 2010; Fisher et al. 2007; Zhang et al. 2006] or placed automatically [Bommes et al. 2009; Knöppel et al. 2013; Panozzo et al. 2014; Ray et al. 2009; Viertel and Osting 2019]. Specialized algorithms exist for enforcing rotational symmetry of the fields [Diamanti et al. 2014; Hertzmann and Zorin 2000; Palacios and Zhang 2007], including six-fold symmetry [Nieser et al. 2012], and for aligning to surface curvature directions [Campen et al. 2012]. Our approach shares many ideas and elements with this long tradition of work, though no existing approach is directly applicable to our problem.

**Discrete Geodesics Field Design.** Pottmann et al [2010] describe three methods for designing geodesic fields on surfaces: (1) sweeping one geodesic into a family of parallel curves along a Jacobi field; (2) a level set method for solving the geodesic equation; and (3) projecting an initial field onto a small basis of approximately-geodesic fields, and “sharpening” into locally-geodesic patches. We compare our approaches to geodesic field design in detail in Section 4.1; briefly, their work is most concerned with finding geodesic vector fields that are locally well-spaced and nearly parallel, but do not thoroughly treat singularities or complex global geometry. Note that the geodesic field design problem is unrelated to that of computing single-source geodesic distances, or specific length-minimizing geodesics between points [Crane et al. 2013; Lin et al. 2013; Mitchell et al. 1987; Polthier and Schmies 2006; Surazhsky et al. 2005].

**Integrating Vector Fields.** There are two classes of methods for robust reconstruction of integral curves from a vector field: one can trace curves by locally integrating the vector field from face to face [Bhatia et al. 2011], and with care numerical integration issues such as intersection of streamlines [Ray and Sokolov 2014] can be avoided. However, tracing methods are fundamentally local, and cannot ensure that streamlines reconnect properly after circulating around holes. Alternatively, integral curve extraction can be viewed as a field-guided mesh parameterization problem, where integral curves become isolines of the parameterization function, as described by Ray et al [2006] and numerous recent papers. Much work has been done on quadrilateral or hexagonal remeshing aligned to 2- or 3-direction fields in particular [Bommes et al. 2009; Campen and Kobbelt 2014b; Jakob et al. 2015], including by representing  $m$ -vector fields by complex polynomials rather than

angles or explicit sets of tangent vectors [Diamanti et al. 2014, 2015]. Parameterization is global, and yields closed loops even around complex topological features, but the orientation of the extracted curves may not match the input vector field if that field is not globally integrable. Although some methods include constraints or objective terms [Campen et al. 2016] to improve alignment, we present a method designed from the ground up to preserve, as much as possible, integral curve orientation.

*Designing Weave Patterns.* Several papers look at computational design of biaxial and triaxial weave patterns. A combinatorial half-edge twisting operation can be used to plain-weave any surface [Akleman et al. 2009], a technique that was demonstrated in practice using paper sculpture [Xing et al. 2011]. The Rhino plugins CompuWoven [Tao et al. 2016] and WeaveMesh [Tao et al. 2017] present users with a simple parametric model of plain weaves, which can be used to explore the space of woven 3D shapes. Campen et al. [2014a] consider the biaxial weaving problem, though they focus primarily on meshing applications. Akleman et al. [2011] describe a method for triaxially twill-weaving any surface, though their method is combinatorial and limited to subsets of edges of a provided input mesh. Similarly, Ayres et al. [2018] describe synthesizing triaxial weaves subordinate to a given triangulation. Takezawa et al. [2016] decompose arbitrary surfaces into two orthogonal families of woven strips aligned to curvature directions. Zwierzycki et al. [2017] explore design and simulation of braided yarn. Garg et al. [2014] compute approximations of surfaces using Chebyshev nets. Also related are methods not directly concerned with weaving, but which optimize approximations of surfaces by interlocking wire loops [Miguel et al. 2016], connected curves [Zehnder et al. 2016], stripe patterns [Ercolani and Venkataramani 2009; Knöppel et al. 2015], mesh joinery [Cignoni et al. 2014], interlocking flexible quadrilateral elements [Skouras et al. 2015], and reciprocal frame structures (stable assemblies of rods in closed circuit) [Song et al. 2013].

*Simulating Rods and Ribbons.* To model ribbon physics in our weave design tool, we adopt the Cosserat rod model of Bergou et al [2010; 2008], as its representation using centerline vertices and edge directors allows us to easily convert integral curves on manifold surfaces into rods. We note in passing that several other approaches to rod physics are available, including models based on masses and springs, super-helices [Bertails et al. 2006], other centerline formulations [Shen et al. 2015], etc. Finally, several papers have studied simulation of knitted cloth, using various reduced representations of the fibers and contacts between them [Cirio et al. 2014; Kaldor et al. 2008; Yuksel et al. 2012].

*Designing Knitted Cloth.* Several recent works describe algorithms for designing knitting patterns that reproduce 3D shapes [Igarashi et al. 2008; Narayanan et al. 2018; Popescu et al. 2018; Wu et al. 2018], modeling or designing knitted cloth [Leaf et al. 2018; Yuksel et al. 2012], etc. Although weaving and knitting are similar in that they both entail fabricating 2D structures from 1D elements, the physical and geometric characteristics of ribbons are very different than that of yarns—as discussed in Section 1, ribbons must be laid flat along a weave, and cannot be deformed into arbitrary stitch patterns.

### 3 PRELIMINARIES AND OVERVIEW

The remainder of the paper is organized into two parts: in the first (Section 4) we will define, discretize, and present algorithms for designing geodesic foliations on curved surfaces. In the second (Section 5), we will use these algorithms to develop a computational pipeline that takes as input a curved surface and computes a weaving pattern that approximates the surface.

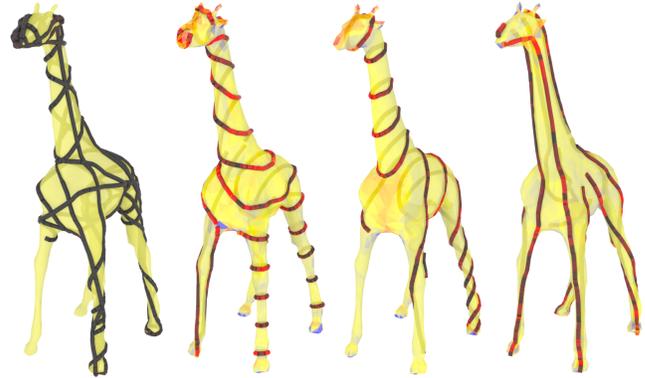


Fig. 3. A qualitative comparison of approaches to designing geodesic giraffe foliations. Naive tracing of intrinsic geodesics over the surface produces a mess of intersecting streamlines (left). Integral curves of as-smooth-as-possible vector fields (center-left) are evenly spaced, but may have high amounts of geodesic curvature, visualized in red. Pottmann et al.'s [2010] spectral design and sharpening method (center-right) produces a locally-geodesic field, but with a large number of singular lines. Our approach (right) yields globally, nearly geodesic foliations.

*Geodesic Foliations.* On a smooth surface  $M$ , a *geodesic foliation* is a one-dimensional foliation of  $M$  whose leaves are all geodesics. This foliation can be represented in two ways, and we will make use of both: (i) as a submersion  $\theta : M \rightarrow S^1$ , whose level sets are immersed closed curves by the inverse function theorem. If the level sets of this periodic function  $\theta$  are all geodesics, then the foliation as a whole is called geodesic. We can also represent geodesic foliations as (ii) complete vector fields  $\mathbf{w}$  on  $M$  whose integral curves are closed geodesics. Examples include any constant vector field on the plane, and vector fields  $\mathbf{w} = \alpha\partial_\theta + \beta\partial_z$  on the cylinder, for constants  $\alpha$  and  $\beta$  (see figure 4). Notice that the two representations are related by  $\mathbf{w} = (\nabla\theta)^\perp$ , where  $\mathbf{v}^\perp$  denotes clockwise rotation of each vector of the vector field  $\mathbf{v}$  by ninety degrees about the surface normal.

Unfortunately, geodesic foliations do not exist on most manifolds [Yampolsky 2005]. However, there is an extremely rich space of *almost*-foliations, which we define equivalently as:

- geodesic foliations of  $M \setminus \mathcal{S}$ , where  $\mathcal{S}$  is a measure-zero set of singularities;
- periodic functions  $\theta : M \rightarrow S^1$  with singularities at  $\mathcal{S}$ , all of whose level sets are geodesics;
- complete vector fields  $\mathbf{w}$  vanishing at  $\mathcal{S}$  whose integral curves are all (not necessarily closed) geodesics.

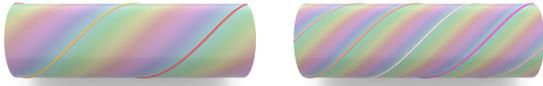


Fig. 4. Two different periodic ( $S^1$ -valued) scalar fields  $\theta_1, \theta_2$  on the surface of the cylinder, and level sets of each. The functions are minimizers of Equation (1), with the only change being to the global rescaling of the normalization constant  $c$ . All of the traced curves are intrinsic geodesics.

To see that this space is rich, notice that for any point  $\mathbf{p} \in M$ , one can take  $S$  to be the cut locus of  $\mathbf{p}$ , with minimizing geodesics to  $\mathbf{p}$  partitioning the rest of  $M$  into a *geodesic almost-foliation*.

Our ultimate goal in Section 4 will be to recover a discrete analogue of the periodic function  $\theta$  defining a geodesic almost-foliation on a triangle mesh  $\mathcal{T}$ . This formulation is convenient, because given  $\theta$  we can then sample a sparse, evenly-spaced collection of level sets of  $\theta$  to guide placement of ribbons when we design weave patterns in Part 2 of our paper. However, to find such a  $\theta$  we will instead first work with the definition based on  $\mathbf{w}$ . From there, we will relax the problem to that of finding *approximately*-geodesic almost-foliations, and present two numerical algorithms: the first computes an approximately-geodesic direction field  $\hat{\mathbf{w}}$  on the faces of  $\mathcal{T}$  whose integral curves are the leaves of the foliation (Equation (5)). The second solves a vector field integration problem to find the  $\theta$  whose isolines align with the integral curves of  $\hat{\mathbf{w}}$  as closely as possible:

$$\min_{\theta, \mathbf{s}} \|\nabla\theta - \mathbf{s}\hat{\mathbf{w}}^\perp\|^2 \quad \text{s.t.} \quad \|\mathbf{s}\|^2 = c, \quad (1)$$

where we discretize  $\theta$  as an  $S^1$ -valued function on the vertices of  $\mathcal{T}$ ,  $\mathbf{s}$  is a scalar field of integration factors on the faces of  $\mathcal{T}$  and  $\mathbf{s}\hat{\mathbf{w}}$  is face-wise multiplication, and  $c$  is a global scale controlling the frequency of  $\theta$ . This approach to splitting computation of  $\hat{\mathbf{w}}$  and  $\theta$  allows us to separate the concerns of finding solutions which are consistent with the *local* surface geometry, and the *global* manifold topology, a common strategy also leveraged in related problems such as surface parameterization. We evaluate our algorithms against prior methods for designing geodesic foliations (see Figures 3 and 7).

*Computational Basket-Weaving.* As we argue in the introduction, weaving surfaces requires laying out the ribbons of the weave along surface geodesics. In Section 5 we apply our geodesic foliation toolbox to design of triaxial weaves approximating arbitrary input surfaces. We will discuss (i) how a triaxial weave can be represented by a geodesic foliation of a six-fold branched cover of the surface, (ii) how to extract a weave pattern using the algorithms from Part 1 of the paper, (iii) how to polish the weave design using elastic ribbon simulation, and (iv) practical aspects of fabricating the weave from real materials. We validate our approach on a variety of fabricated and simulated examples.

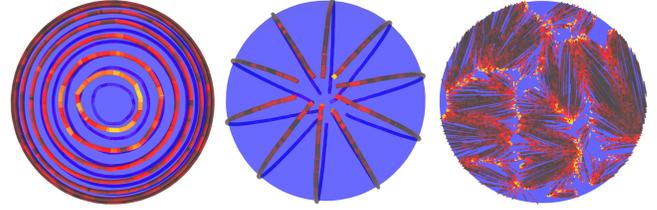


Fig. 5. Directly minimizing the Dirichlet energy of a vector field on a manifold leads to singularities with vorticity (integral curves traced on *left*). Our energy produces singularities that are irrotational and recovers geodesics with non-uniform spacing (*center*). Further, our method can recover fields which are almost everywhere geodesic except on glassy “domain wall” [Aharoni et al. 2017] boundaries by conditioning with very small amounts of global smoothness ( $\lambda = 10^{-8}$  here) (*right*).

## 4 PART 1: DISCRETE GEODESIC FOLIATIONS

In this part, we discuss how to solve for approximately-geodesic almost-foliations of discrete two-manifolds (represented as triangle meshes  $\mathcal{T}$ ). As discussed in Section 3, there are two ways to represent the leaves of such foliations: as level sets of a function  $\theta : \mathcal{T} \rightarrow S^1$ , or as integral curves of a vector field  $\mathbf{w}$  on  $\mathcal{T}$ . Lacking an obvious way to solve for  $\theta$  directly, we proceed in two steps: first, we describe an algorithm for finding direction fields  $\hat{\mathbf{w}}$  whose integral curves are nearly geodesics (Section 4.1). Second, we describe a global integration algorithm (Section 4.2) that recovers a  $\theta$  whose isolines are well-aligned to a given input direction field; this algorithm is general, and when applied to the geodesic direction field  $\hat{\mathbf{w}}$ , computes a  $\theta : \mathcal{T} \rightarrow S^1$  whose isolines are approximately geodesic.

### 4.1 Discrete Geodesic Field Design

In the discrete setting, it is straightforward to define geodesic *curves*, either as arclength-minimizing paths between pairs of points, or integral curves of the gradient of the geodesic distance function from a source point; several algorithms efficiently compute geodesic distance and geodesic curves [Crane et al. 2013; Mitchell et al. 1987; Polthier and Schmieß 2006; Surazhsky et al. 2005]. We instead want an entire unit *vector field*  $\hat{\mathbf{w}}$  whose integral curves are all discrete geodesics, where we make the common choice to represent  $\hat{\mathbf{w}}$  as an assignment of tangent vectors  $\hat{\mathbf{w}}_i$  to the faces  $\mathcal{F}$  of  $\mathcal{T}$  [de Goes et al. 2016]. It is not obvious how to discretize the geodesic condition to entire vector fields. We will present our definition below, in Section 4.1.1, but first discuss some less suitable approaches, which motivated our choices.

*Via Geodesic Integral Curves.* One might ask for all of  $\hat{\mathbf{w}}$ ’s integral curves, when traced over  $\mathcal{T}$ , to be discrete geodesics; unfortunately there are no such fields except on intrinsically flat  $\mathcal{T}$ . Denote by  $T_{j \leftarrow i}$  the parallel transport operator rotating tangent vectors on face  $i$  to those on neighboring face  $j$  by “unfolding” face  $i$  along their common edge<sup>1</sup>. To have geodesic integral curves the field would

<sup>1</sup>This operator  $T_{j \leftarrow i}$  can be viewed as a discrete affine connection, and is trivial, but should not be confused with the more sophisticated “trivial connections” of Crane et al [2010], which also include an in-plane rotation by a correction angle in order to concentrate singularities at a few user-specified points.

need to satisfy

$$\angle(\mathbf{w}_j, T_{j \leftarrow i} \mathbf{w}_i) = 0 \quad (2)$$

for every pair of neighboring faces  $i, j$ , which is impossible since around any interior vertex with nonzero angle deficit, the  $T$ s do not compose to the identity [Crane et al. 2010].

*Via the Geodesic Equations.* A different task is to discretize geodesic fields by way of the geodesic equation  $\nabla_{\hat{\mathbf{w}}} \hat{\mathbf{w}} = 0$ . The chief difficulty here is that on surfaces of genus  $\neq 1$ , singularities are inevitable, and at such singularities the covariant derivative  $\nabla \hat{\mathbf{w}}$  of the field is not well-defined. Although there has been some work studying discrete covariant derivative operators [Azencot et al. 2015; Liu et al. 2016], defining discrete geodesic fields so that they are well-behaved near singularities requires a discrete notion of covariant derivative that is insensitive to them, motivating the following.

*Via Curl and Unit-norm Conditions.* Since  $\hat{\mathbf{w}}$  is unit, the geodesic equation is equivalent to vanishing of the field’s curl,  $\nabla \times \hat{\mathbf{w}} = 0$ . This relationship is at the heart of Pottmann et al [2010]’s third geodesic design algorithm; they defined a discrete curl in terms of failure of the Jacobian averaged over triangle neighborhoods to be symmetric, which is again ill-defined near singularities. A singularity-insensitive characterization of curl-free discrete vector fields has been studied in the context of vector field integration [Diamanti et al. 2015; Polthier and Preuß 2003]. Let  $\mathcal{E}$  denote the edge set of  $\mathcal{T}$  and  $\mathcal{E}_{\text{int}}$  the interior edges, with  $\mathbf{e}_{ij}$  the edge common to neighboring faces  $i$  and  $j$  (and oriented consistently with face  $i$ ). A vector field  $\mathbf{v}$  is curl-free if, for every edge,

$$\mathbf{v}_i \cdot \mathbf{e}_{ij} = \mathbf{v}_j \cdot \mathbf{e}_{ij}, \quad (3)$$

which we can write compactly as  $C\mathbf{v} = \mathbf{0}$  for linear curl operator  $C_{|\mathcal{E}_{\text{int}}| \times 2|\mathcal{F}|}$ . This curl-free condition is well-defined even near singularities (see Figure 6, left).

Unfortunately, the condition  $C\hat{\mathbf{w}} = \mathbf{0}$  together with the unit-norm condition  $\|\hat{\mathbf{w}}_i\| = 1$  over-constrains the space of geodesic vector fields, as can be seen by counting degrees of freedom (the vector on each face is parameterized by an angle, but constrained across each edge), or by noting that the choice of a unit vector on one face eliminates all but two choices of unit vector on each neighboring face. Most non-flat triangle meshes  $\mathcal{T}$  admit no vector fields that are both curl-free and unit, in contrast to the rich space of geodesic almost-foliations in the smooth setting.

**4.1.1 Discrete Geodesic Fields.** Starting from the expression of the geodesic equation in terms of curl, we relax the unit-norm constraint on the field and define a discrete geodesic field  $\hat{\mathbf{w}}$  as any solution to

$$\arg \min_{\hat{\mathbf{w}}, \delta} \frac{1}{2} \|\delta\|^2 \quad \text{s.t.} \quad \begin{aligned} C(\hat{\mathbf{w}} + \delta) &= \mathbf{0} \\ \|\hat{\mathbf{w}}_i\| &= 1. \end{aligned} \quad (4)$$

Notice that at critical points of Equation (4), the field  $\hat{\mathbf{w}}$  and the residual  $\delta$  must be everywhere parallel<sup>2</sup>. In the smooth setting, the solutions are geodesic fields, with  $\delta = 0$  (see appendix A). In the discrete setting, the intersection of the space of curl-free fields and that of unit fields is insufficiently rich; optimal solutions to equation (4) are those where simultaneously  $\hat{\mathbf{w}} + \delta$  is as unit as

<sup>2</sup>Since otherwise, the objective value can be improved by setting  $\hat{\mathbf{w}} \leftarrow \frac{\hat{\mathbf{w}} + \delta}{\|\hat{\mathbf{w}} + \delta\|}$ .

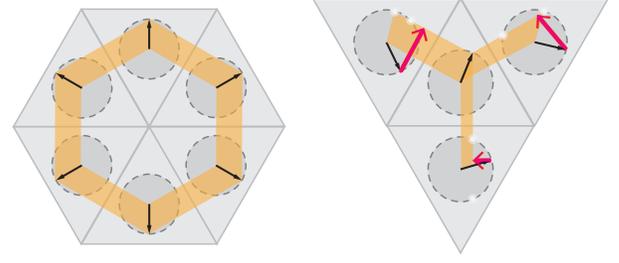


Fig. 6. *Left:* discrete curl measures whether vectors on neighboring faces agree upon projection onto the common edge, and is well-defined near singularities. *Right:* unfortunately, requiring a vector field  $\hat{\mathbf{w}}$  to be both unit-magnitude and discretely curl-free overconstrains the field. Instead, given a unit vector field  $\hat{\mathbf{w}}$  (black vectors), our algorithm finds the minimal perturbation  $\delta$  (red vectors) needed so that  $\hat{\mathbf{w}} + \delta$  is curl-free. Our relaxed definition of a discrete geodesic field is one for which  $\delta$  is locally minimized.

possible, while  $\hat{\mathbf{w}}$  is as curl-free as possible, with  $\|\delta\|$  quantifying the failure of these conditions in both cases (see Figure 6, right).

The above variational characterization of discrete geodesic fields can be used for geodesic field design, by choosing an initial unit field  $\hat{\mathbf{w}}^0$  and flowing it towards optimality. As we will discuss below, despite its non-convexity the variational problem can be solved with guaranteed convergence to a stationary point. Note, though, that the solutions of Equation 4 may contain singularities. This property is necessary and desirable, but choosing  $\hat{\mathbf{w}}^0$  arbitrarily can yield geodesic fields that are “glassy” with an undesirably large number of singularities. For design applications, it is useful to promote smoothness of the field, which we address in the next section and illustrate in Figures 9 and 10.

**4.1.2 Smoothness and Control.** A geodesic vector field can have arbitrarily many singularities, so for design it is useful to (1) tune between field smoothness and geodesicness; and (2) force alignment of the field to user-prescribed handles. Both can be easily incorporated into the variational problem in Equation (4). Let  $\mathcal{H} \subset \mathcal{F}$  denote a set of faces on which the user wishes to constrain the vector field to some predetermined directions  $\hat{\mathbf{w}}^{\mathcal{H}}$ ; then an approximately-geodesic field obeying the handle constraints can be computed by solving

$$\begin{aligned} C(\hat{\mathbf{w}} + \delta) &= \mathbf{0} \\ \|\hat{\mathbf{w}}_i\| &= 1 \\ \arg \min_{\hat{\mathbf{w}}, \delta} \frac{1}{2} \|\delta\|^2 + \frac{\lambda}{2} \|\nabla(\hat{\mathbf{w}} + \delta)\|^2 \quad \text{s.t.} \quad & \hat{\mathbf{w}}_i = \hat{\mathbf{w}}_i^{\mathcal{H}} \quad \forall i \in \mathcal{H} \\ & \delta_i = 0 \quad \forall i \in \mathcal{H}, \end{aligned} \quad (5)$$

where the parameter  $\lambda$  controls the smoothness of the field. As  $\lambda$  grows large, Equation (5) prioritizes smoothness over straightness of the field, and penalizes singularities<sup>3</sup>; Figure 9 shows the effect of  $\lambda$  when optimizing equation (5) for initially-random fields  $\hat{\mathbf{w}}^0$  on the unit disk. Discrete geodesic fields can be designed by first

<sup>3</sup>We note in passing that the variational problem (5) bears similarity to *Aviles-Giga functionals* originally studied in mathematical physics to better understand magnetization in thin films [1987], and later discretized [Jabin et al. 2002; Riviere and Serfaty 2001], though these functionals place an additional constraint on the divergence of  $\delta$ .

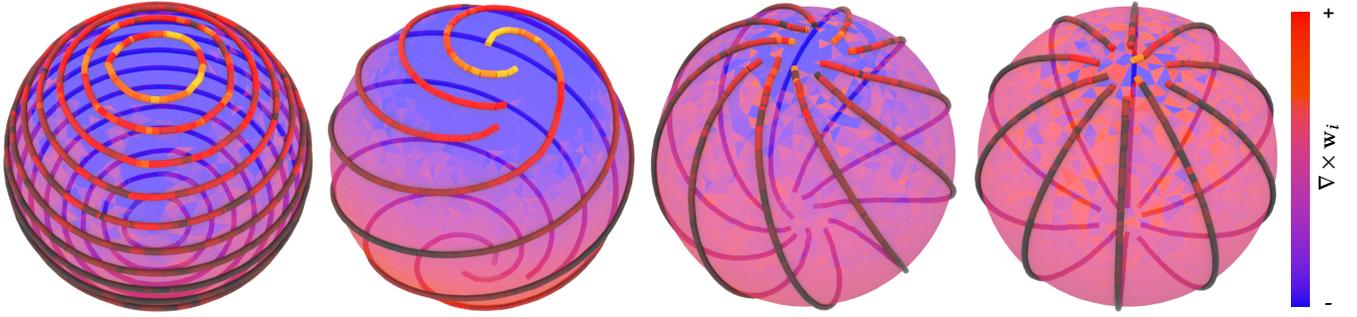


Fig. 7. Streamlines of a vector field on a discretized unit sphere, where the field was optimized using (from left to right): Knöppel et al. [2013]’s Dirichlet energy formulation; Knöppel et al.’s anti-holomorphic energy formulation; Pottmann et al. [2010]’s field optimization; our geodesic field optimization. Notice that only our method yields a field aligned to the expected great circles, with no vorticity at the two field singularities. Colors of streamlines encode geodesic curvature (warmer curves have higher curvature). Colormap on the surface (right) indicates the signed curl, as calculated in Appendix B.1.

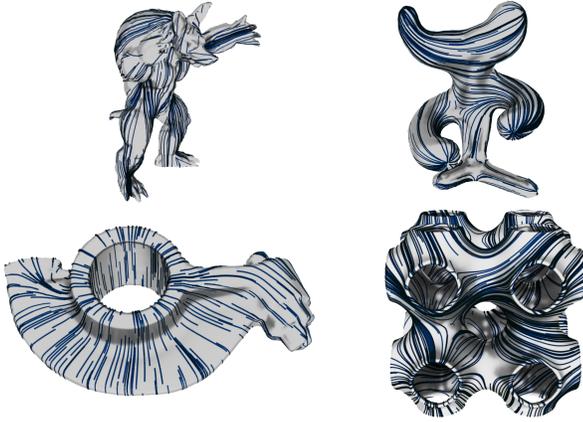


Fig. 8. We compute geodesic fields on a variety of surfaces with complex geometry and topology. We include a numerical evaluation in Table 1

solving for  $\hat{\mathbf{w}}$  with  $\lambda$  large, then “sharpening” [Pottmann et al. 2010] the field by reducing  $\lambda$ ; all geodesic fields we show in this paper were designed using this methodology.

Figures 3, 7, and 8 show the results of our method on a variety of surface geometries and topologies. On simple shapes, such as the sphere or disk, our method reproduces fields whose streamlines are the expected straight lines and great circles. On more complex shapes, our method produces fields whose integral curves are locally geodesics, with no vorticity near singularities.

*Relation to Pottmann et al.* Pottmann et al. [2010] propose several algorithms for designing geodesic fields. Their level set method is purely local and cannot be applied to surfaces with non-disk topology. Their sharpening-based pipeline does succeed in computing high-quality geodesic fields on local patches, but it is not designed to be robust near singularities and does not yield vorticity-free fields there; see, for instance, Figure 7.

**4.1.3 Numerical Solution.** Although Equation (5) is not convex and admits multiple solutions, it can be solved in alternating fashion, by solving a linear least squares problem to update  $\delta$ , and then

Table 1. Curl statistics for vector fields optimized using Algorithm 1, for the fields shown in Figure 8. We initialize each field  $\hat{\mathbf{w}}^0$  by removing the curl constraint in Equation 5, which produces smooth vector fields qualitatively similar to the globally optimal fields of Knöppel et al. [2013]. The left column lists the total curl (measured using the methodology described in Appendix B.1) of this initial field, and the right, the curl of optimized field. Our algorithm significantly reduced the total curl (and hence the geodesicness of the field’s integral curves).

| Model        | Pre-optimization Curl | Post-optimization Curl |
|--------------|-----------------------|------------------------|
| Armadillo    | 20.60                 | 7.32                   |
| Moomoo       | 13.36                 | 4.19                   |
| Rocker Arm   | 8.78                  | 6.6                    |
| Schwarz- $p$ | 30.15                 | 15.65                  |

directly updating  $\hat{\mathbf{w}}$  to align with  $\hat{\mathbf{w}} + \delta$ , as described in Algorithm 1. Notice that this greedy update of  $\hat{\mathbf{w}}$  is also optimal, given fixed  $\delta$ . Each step of the alternation is guaranteed to decrease the objective function while maintaining all constraints, so convergence to a stationary point is assured. Appendix B contains more details about the algorithm, including implementation hints and convergence proof.

**ALGORITHM 1:** Computing approximate discrete geodesic fields given an initial guess  $\hat{\mathbf{w}}^0$  and handle constraints.

```

for  $i = 1, \dots, i_{max}$  do
     $\tilde{\delta} \leftarrow \arg \min_{\delta} \frac{1}{2} \|\delta\|^2 + \frac{\lambda}{2} \|\nabla(\hat{\mathbf{w}}^{i-1} + \delta)\|^2$  s.t.  $C(\hat{\mathbf{w}}^{i-1} + \delta) = \mathbf{0}$  ;
     $\delta_j = 0 \forall j \in \mathcal{H}$  ;
    for  $j = 1, \dots, |\mathcal{F}|$  do
         $\hat{\mathbf{w}}_j^i \leftarrow (\hat{\mathbf{w}}_j^{i-1} + \tilde{\delta}_j) / \|\hat{\mathbf{w}}_j^{i-1} + \tilde{\delta}_j\|$ ;
         $\delta_j^i \leftarrow \hat{\mathbf{w}}_j^{i-1} + \tilde{\delta}_j - \hat{\mathbf{w}}_j^i$ ;
    end
    if  $\|\hat{\mathbf{w}}^i - \hat{\mathbf{w}}^{i-1}\| < \epsilon$  then break;
end

```

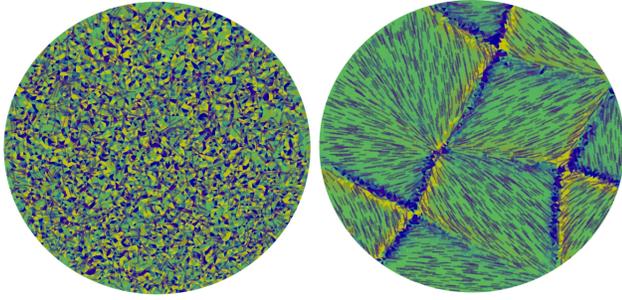


Fig. 9. The parameter  $\lambda$  in equation 5 gives us extra control over our vector field design process. Initializing a random vector field on the unit disk and setting  $\lambda = 0$  produces very disordered solutions (left). Setting  $\lambda = 10^{-7}$  crystallizes the field into large ordered domains (right), and increasing  $\lambda$  further results in a uniform field. The heatmap illustrates signed curl.

**4.1.4 Sharpening Incompatible Handles.** We can probe the behavior of Equation 5 by assigning handles  $\mathcal{H}$  that admit no smooth global geodesic vector field, and running Algorithm 1 for different choices of  $\lambda$  (see Figure 10). In the limit  $\lambda \rightarrow \infty$ , Equation 5 converges to the convex problem of computing the smoothest-possible curl-free field on  $\mathcal{T}$ ; this field is far from unit-magnitude (and so far from being geodesic) and  $\delta$  is large throughout the surface (left). As  $\lambda \rightarrow 0$ , solutions qualitatively change: the field becomes geodesic away from sharp domain wall boundaries (top right) and  $\delta$  becomes sparse, away from point and line dislocations (bottom right).

*Relation to Smoothness Optimization.* Many techniques exist for optimizing vector or direction fields with respect to different smoothness criteria, and subject to design constraints. Most methods are tailored to the quadratic Dirichlet energy  $\frac{1}{2}\|\nabla\mathbf{w}\|^2$ , or its generalizations, such as the Killing [Ben-Chen et al. 2010] or so-called (anti-)holomorphic energies [Knöppel et al. 2013]; connections have also been drawn [Viertel and Osting 2019] to the Ginzburg-Landau functional from mathematical physics. While smoothness energy measures also promote “straightness” of a vector field in some sense, and many of these methods are convex and straightforward to solve (using e.g. a sparse eigenvalue problem [Knöppel et al. 2013], gradient descent via LBFGS, approximation via MBO [Viertel and Osting 2019]) smoothness is not equivalent to the (nonlinear) geodesic energy: for instance, consider that the vector field  $\mathbf{w}(x, y) = (x, y)$  is geodesic on any neighborhood of the punctured plane, but has non-vanishing Dirichlet energy. Note that all quadratic energies share the same shortcoming: near singularities, the vector field vanishes, and so does the energy density, regardless of the vorticity of the field. Figure 3 compares our method to fields that minimize Dirichlet or antiholomorphic energy.

## 4.2 Recovering $\theta$

The geodesic field  $\hat{\mathbf{w}}$  on  $\mathcal{T}$  computed by Algorithm 1 in principle represents a discrete geodesic almost-foliation. However, in practice one desires to extract a sparse set of leaves of the foliation, so that these curves are nicely-spaced over the surface. An implicit representation of the geodesic foliation in terms of a function  $\theta$ :

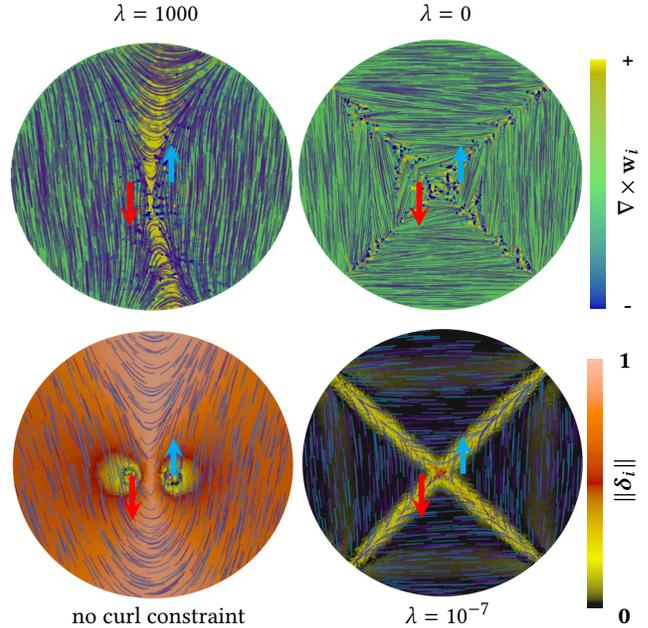


Fig. 10. We compute solutions to Equation 5 on a disk with two handles (red and cyan arrows), and  $\lambda = 10^3$  (left) and  $\lambda \rightarrow 0$  (right). We visualize the curl of the resulting field  $\hat{\mathbf{w}}$  (top) and magnitude of  $\delta$  (bottom). As the smoothness regularization  $\lambda$  sharpens to zero, curl and  $\delta$  concentrates at singularities, globally improving geodesicness of  $\hat{\mathbf{w}}$ . Without the curl constraint the  $\delta$  distribution is similar to the one in the  $\lambda \rightarrow \infty$  limit, but the underlying fields are qualitatively different (purple traces, left).

$\mathcal{T} \rightarrow S^1$  is more useful for this purpose, since standard isoline-extraction algorithms can then be applied to  $\theta$ . In this section, we discuss the challenges in recovering such a  $\theta$  from a discrete geodesic field, and present a new algorithm for doing so.

**4.2.1 Vector Field Integrability.** Let us first examine the problem of finding  $\theta$  in the smooth setting, for a unit vector field  $\hat{\mathbf{w}}$  on an oriented manifold  $M$ . In order for  $\theta$ 's isocontours to agree with integral curves of  $\hat{\mathbf{w}}$ , the gradient of  $\theta$  must lie parallel to  $\hat{\mathbf{w}}^\perp$  at all point  $\mathbf{p} \in M$ :

$$\nabla\theta(\mathbf{p}) = s(\mathbf{p})\hat{\mathbf{w}}(\mathbf{p})^\perp, \quad s(\mathbf{p}) > 0. \quad (6)$$

Notice that such a  $\theta$  is only expected to exist if we rescale the input field  $\hat{\mathbf{w}}$  by a scalar field of integration factors  $s : M \rightarrow \mathbb{R}^+$ ; for otherwise  $\hat{\mathbf{w}}$  and  $\hat{\mathbf{w}}^\perp$  are both curl-free, and therefore  $\hat{\mathbf{w}}$  is unit-norm and harmonic, impossible except on surfaces of constant curvature [Boeckx and Vanhecke 2000].

The trick then is finding an  $s$  for which such a  $\theta$  exists.  $s\hat{\mathbf{w}}^\perp$  cannot be the gradient of a scalar function unless it is *globally integrable*: for each closed curve  $\gamma : S^1 \rightarrow M$ , we must have  $\oint_\gamma \langle s\hat{\mathbf{w}}^\perp, \gamma' \rangle ds = 0$ . If  $M$  is simply-connected, it is enough to check that  $s\hat{\mathbf{w}}^\perp$  is *locally integrable*, i.e. is curl-free; more generally, it is sufficient for  $s\hat{\mathbf{w}}^\perp$  to be curl-free, and for the global integrability condition to be satisfied for all generators  $\gamma$  of the fundamental group of  $M$ .

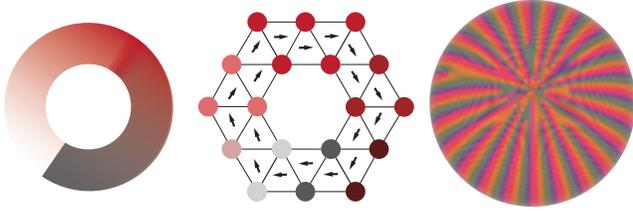


Fig. 11. Rescaling a vector field to be locally integrable isn't sufficient to guarantee global integrability. Consider, for instance, an annulus (left) around which any  $S^1$ -valued function  $\theta$  must have period  $2\pi N$ . We discretize such functions on the vertices of a triangle mesh (center) and given a vector field, find the  $\theta$  whose gradient best matches a rescaling of the field. Mesh resolution limits the frequency of  $\theta$ , and we find solutions with a sparse collection of edge dislocation [Knöppel et al. 2015], as illustrated for a geodesic field on a sphere (right).

Given  $\hat{w}^\perp$ , does there exist a rescaling  $s$  of the vector field which is locally integrable? Globally integrable? Both questions are surprisingly subtle. Unfortunately, existence of an  $s$  is not guaranteed, even for smooth  $\hat{w}^\perp$  on a simply-connected neighborhood of the plane [Qmechanic 2018]; one counterexample [Boyling 1968] is

$$w^\perp(x, y) = (y^3(1-y)^2, y^3 - 2(1-y^2)),$$

which is not integrable on any neighborhood of a point on the line  $y = 1$  (integrability would require  $s$  to diverge as  $y \rightarrow 1$ ).

Global integrability poses its own challenges. The codomain of  $\theta$  was chosen to be  $S^1$  rather than  $\mathbb{R}$  since it is clear that  $\theta$  must be periodic when  $w$  wraps around holes in  $M$  or singularities in the field (see Figures 4 and 11). Methods in the literature allow periodicity of  $\theta$  either by introducing integer variables that encode jumps in  $\theta$  on the universal cover of  $M$  [Bommes et al. 2009], or by representing  $\theta$  with trigonometric functions or complex numbers [Knöppel et al. 2015; Ray et al. 2006]. Note that allowing periodic  $\theta$  is necessary, but still insufficient, to guarantee existence of  $s$ , even for locally-integrable vector fields.

*Our Approach.* We do not attempt a full solution of the (ill-posed) global vector field integration problem. Rather, we present a new variational approach for finding a good  $s$  which renders  $s\hat{w}^\perp$  as close as possible to being integrable, based on the same relaxation of discrete vector field integrability we leveraged in Section 4.1. With a good estimate for  $s$  we can apply prior work on global integration [Knöppel et al. 2015; Ray et al. 2006] to round  $s\hat{w}^\perp$  to a globally integrable field without significantly altering the orientation of its streamlines. We show significant improvement of this approach over alternative global integration techniques (see Figure 15).

**4.2.2 Preprocessing the Mesh.** Since the vector field  $\hat{w}$  encodes an *almost*-foliation, it can have singularities, and  $\theta$  is only expected to be continuous and smooth away from those singularities. To simplify subsequent calculations, we compute the singularities  $\mathcal{S}$  of  $\hat{w}$  (by computing the index of the vector field when circulating around every vertex of  $\mathcal{T}$ ) and puncture  $\mathcal{T}$  by deleting from it all singular vertices and their neighboring faces. Since  $\theta$  is a submersion

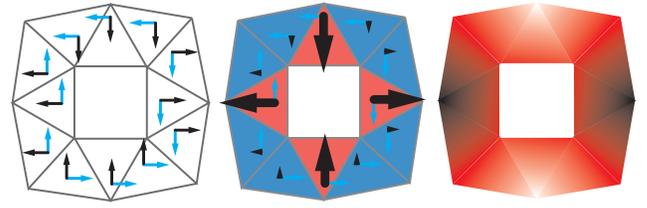


Fig. 12. We seek to recover an  $S^1$ -valued function  $\theta$  on the vertices of the mesh  $\mathcal{T}$  whose isolines are as close as possible to the integral curves of an input unit vector field  $\hat{w}$ . Left: a field  $\hat{w}$  (in cyan) oriented similarly to Figure 10 (top-right), and its rotation  $w^\perp$  (in black). Middle: we solve for a scalar field  $s$  per face so that  $sw^\perp$  (rescaled black vectors) is as close to integrable as possible; here  $s$  vanishes near each line singularity along the four diagonals. Right: we recover  $\theta$ , a vertex-based scalar function whose gradient aligns to  $sw^\perp$  as closely as possible, by solving Equation (8).

on  $\mathcal{T} \setminus \mathcal{S}$ , this preprocessing step allows us to assume  $\theta$  is smooth (and continuous) for all subsequent calculations.

**4.2.3 Approximating an Initial  $s$ .** Recall that a discrete vector field  $v$  on the faces of  $\mathcal{T}$  is locally the gradient of a scalar function on the vertices if and only if it satisfies the edge-based compatibility condition  $Cv = 0$  (see Equation (3)). We thus desire a rescaling  $v_i = s_i \hat{w}_i^\perp$  of  $\hat{w}^\perp$  with  $Cv$  approximately zero. Since per the above discussion, an exact solution is generally not expected to exist, our main idea is to apply the same relaxation as in the geodesic field design step, and consider vector fields of the form  $s\hat{w}^\perp + \delta$ , and ask for the smallest possible  $\delta$  for which this relaxed vector field is integrable. Notice that the equation  $C(s\hat{w}^\perp + \delta) = 0$  is linear in both sets of unknown variables  $s_i$  and  $\delta_i$ ; if we unroll  $\delta$  into a vector, we can write this integrability constraint as

$$B \begin{bmatrix} \delta \\ s \end{bmatrix} = 0$$

for matrix  $B$ , and solve for an optimal  $s$  and  $\delta$  using the variational problem

$$\arg \min_{s, \delta} \frac{1}{2} \|\delta\|^2 + \frac{\mu}{2} \|\nabla s\|^2 \quad \text{s.t.} \quad \begin{bmatrix} B \\ \delta \end{bmatrix} \begin{bmatrix} \delta \\ s \end{bmatrix} = 0, \quad (7) \\ \|s\|^2 + \|\delta\|^2 = 1,$$

where  $\mu$  controls a regularization term promoting smoothness of  $s$ ; we use  $\mu = 10^{-4}$  in all of our examples. The norm constraint is needed to prevent degeneration since the problem is otherwise invariant to uniform scaling of  $\delta$  and  $s$ . The matrix  $B$  is sparse, and Equation 7 can be solved to global optimality as a generalized eigenvalue problem. For details, see Appendix C. Figure 12 illustrates an example of optimizing for  $s$ .

*Relation to Local Curl Minimization.* It is tempting to try directly solving for  $s$  to make  $v^\perp$  as integrable as possible, without use of the  $\delta$  relaxation, e.g. by computing

$$\arg \min_s \frac{1}{2} \left\| B \begin{bmatrix} \delta \\ s \end{bmatrix} \right\|^2 \quad \text{s.t.} \quad \|s\|^2 = 1.$$

This idea has been proposed in work on surface parameterization as a “curl-correcting” preprocessing step; Ray et al. [2006] discretized

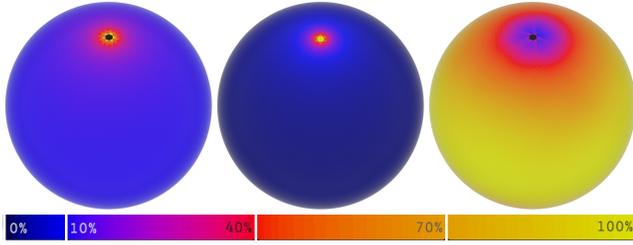


Fig. 13. Using our method, we compute the rescaling field  $s$  needed to make the geodesic field on the unit sphere, shown in Figure 7 left, discretely integrable (left). We compare against the ground truth  $s$  (center), and against  $s$  computed using local curl correction [Ray et al. 2006] (right).

curl using linear finite elements over each triangular face. Although these ideas share with our solution the goal of rescaling  $\hat{\mathbf{w}}^\perp$  to be approximately locally integrable, there is a subtle but important difference: where curl-correction seeks to minimize curl of the vector field on each triangle, our optimization minimizes the global distance of  $s\hat{\mathbf{w}}^\perp$  away from the nearest (pointwise) integrable vector field. Minimizing local measures of curl to find  $s$  suffers from the same pitfall as trying to design geodesic fields by minimizing the  $L_2$  norm of curl: near singularities, it is equally efficient to reduce curl by scaling the vector field, as by straightening it. Our variational approach, equation (7), results in significant improvement in the quality of the rescaled vector field, especially in regions requiring very large rescaling of  $\hat{\mathbf{w}}^\perp$ . We compare against local curl reduction in Figures 13 and 15.

**Globally Scaling  $s$ .** After solving Equation (7) and recovering a rescaling of  $\hat{\mathbf{w}}^\perp$  that is close to locally integrable, there is no guarantee that  $s\hat{\mathbf{w}}^\perp$  is close to *globally* integrable (see Figure 11). To recover  $\theta$ , we use the  $s$  from the above initialization step as an initial guess to an alternating optimization algorithm that jointly finds  $s$  and  $\theta$ . Before this joint optimization, we rescale  $s$  (whose scale was set arbitrarily in Equation (7)) so that  $s\hat{\mathbf{w}}^\perp$  is not too large on any individual edge of  $\mathcal{T}$ ; this rescaling avoids recovery of a  $\theta$  that is aliased on mesh edges<sup>4</sup>. After this automatic rescaling, the user can further multiply  $S$  by an optional scale parameter to control the frequency of  $\theta$ . We make use of this parameter in Part 2 to decrease the ribbon density in our weaving patterns.

**4.2.4 Joint Optimization of  $s$  and  $\theta$ .** The optimization described in the previous section allows us to find a rescaling  $s\hat{\mathbf{w}}^\perp$  that renders  $\hat{\mathbf{w}}^\perp$  approximately locally integrable. We can then apply standard periodic global parameterization techniques to jointly compute  $\theta : \mathcal{T} \rightarrow S^1$  (discretized as an assignment of  $\theta_\alpha \in S^1$  to each vertex  $\alpha$ ) and an update to  $s$ , so that  $\nabla\theta$  well-aligns with  $s\hat{\mathbf{w}}^\perp$ . The basic idea [Knöppel et al. 2015; Ray et al. 2006] is the following: if  $\mathbf{p}_\alpha$  and  $\mathbf{p}_\beta$  are two vertices of face  $i$ , then agreement of  $\nabla\theta$  with  $s\hat{\mathbf{w}}^\perp$  can

<sup>4</sup>Knöppel et al. [2015] propose a solution to aliasing (their “frequency adjustment” step) that takes the advantage of knowing how many periods of  $\theta$  to expect over an edge, given  $s$ . However our aim is to recover sparse integral curves, and so restrict ourselves to solutions with at most half a period per edge. Specifically, we compute

$$\rho = \max_{e_{ij} \in \mathcal{E}} |s_i \hat{\mathbf{w}}_i^\perp \cdot \mathbf{e}_{ij}|$$

and globally scale  $s$  by  $\pi/\rho$ .

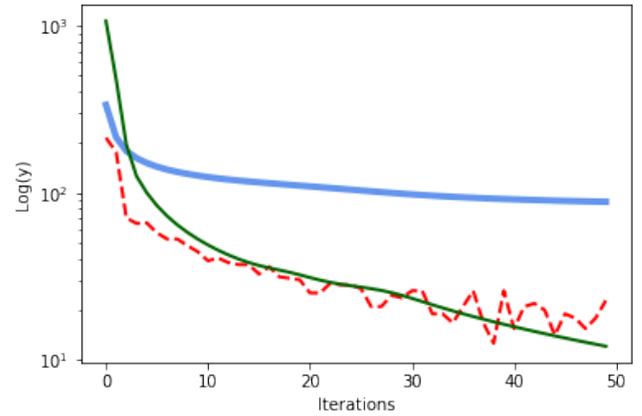


Fig. 14. Convergence of the alternating minimization of equation (8). For the fertility model in figure 15 we show, over 50 iterations, the value of the energy (8) (blue, bold) and the changes  $\|\Delta\theta\|$  (red, dashed) and  $\|\Delta s\|$  (green) to  $\theta$  and  $s$  each iteration.

be written as

$$R(\theta_\beta) = R[\theta_\alpha + s_i \hat{\mathbf{w}}_i^\perp \cdot (\mathbf{p}_\beta - \mathbf{p}_\alpha)],$$

where  $R(\theta) = [\cos \theta, \sin \theta]$ . Joint global recovery of  $\theta$  and  $s$  then amounts to optimizing

$$\min_{\theta, s} \sum_{i \in \mathcal{F}} \sum_{\alpha, \beta} \frac{\omega_{\alpha, \beta}}{2} E_{i, \alpha, \beta}^{\text{iso}}(\theta, s) + \frac{\mu}{2} \|\nabla s\|^2 \quad (8)$$

where the second sum is over consecutive vertices  $\alpha, \beta$  of face  $i$ ,  $\omega_{\alpha, \beta}$  is an inner product weight on edges (we use the cotangent weight of the edge), and

$$E_{i, \alpha, \beta}^{\text{iso}}(\theta, s) = \left( R(\theta_\beta) - R[\theta_\alpha + s_i \hat{\mathbf{w}}_i^\perp \cdot (\mathbf{p}_\beta - \mathbf{p}_\alpha)] \right)^2$$

measures failure of  $\theta$  and  $s\hat{\mathbf{w}}^\perp$  to agree on each edge.

We solve Equation (8) in alternating fashion, by iteratively computing (i) the scale values  $s$  while holding the  $\theta$  fixed, with Gauss-Newton optimization; and (ii) the function  $\theta$ , while holding  $s$  fixed, which amounts to an eigenvector problem [Knöppel et al. 2015]. In all of our examples, we found that ten alternations of steps (i) and (ii) was sufficient. We illustrate the convergence of this procedure in Figure 14.

**4.2.5 Pipeline Summary.** We briefly summarize our full pipeline for finding approximately-geodesic almost-foliations  $\theta$ , given an initial guiding unit vector field  $\hat{\mathbf{w}}$ :

- (1) project  $\hat{\mathbf{w}}$  onto a geodesic vector field, by solving Equation (5) using Algorithm 1;
- (2) puncture  $\mathcal{T}$  at the singularities of  $\hat{\mathbf{w}}$  (Section 4.2.2);
- (3) solve for an initial estimate of  $s$  for which  $s\hat{\mathbf{w}}^\perp$  is close to being discretely integrable (Section 4.2.3);
- (4) globally rescale  $s$  to avoid aliasing across edges;
- (5) refine  $s$ , while solving for  $\theta$ , as described in Section 4.2.4.

Figure 15 shows the results of steps 2–5 of the above pipeline when applied to two examples. Notice that in both cases the isolines of the resulting  $\theta$  are well-aligned to the geodesic field  $\hat{\mathbf{w}}$ , even near

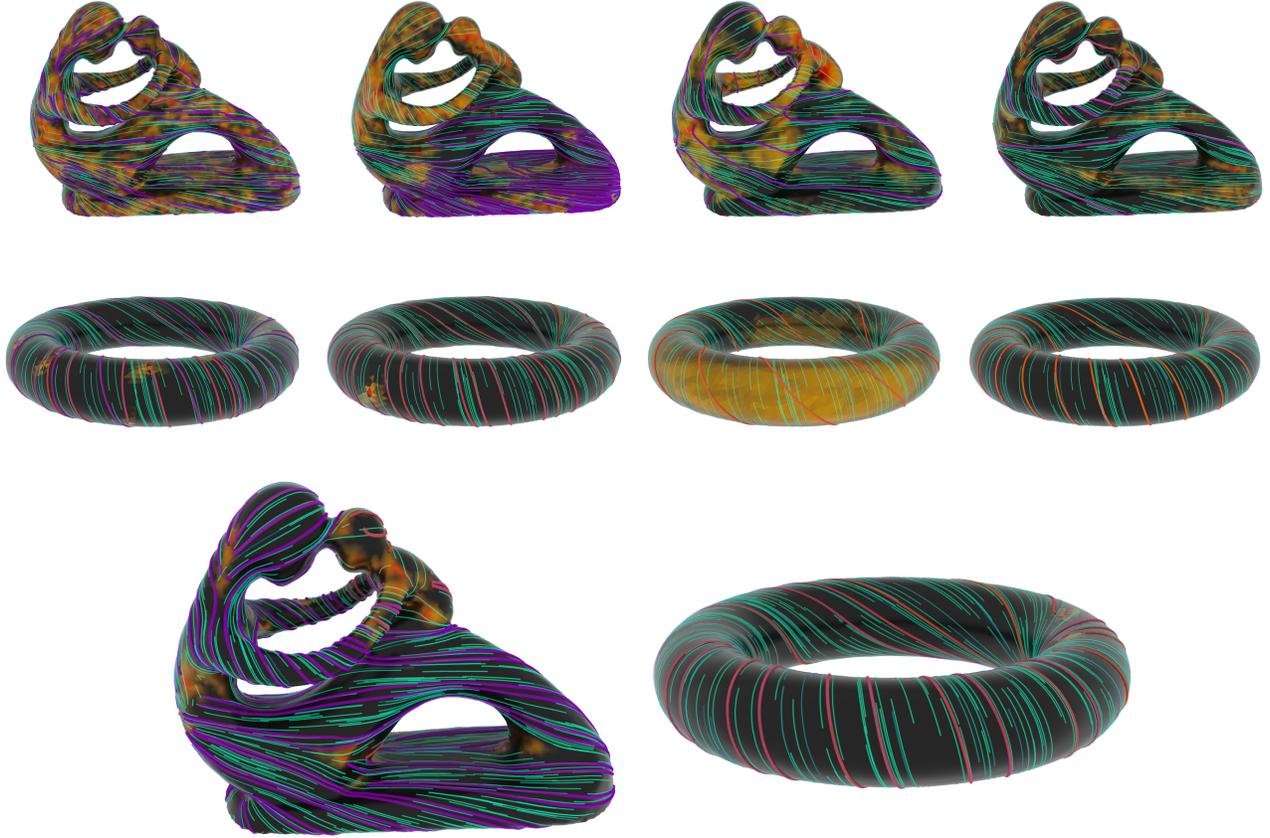


Fig. 15. For fertility and torus, we compare our global integration method with others. For the first two rows, we compute  $\theta$  and extract isolines using, from left to right: (i) PGP-based [Knöppel et al. 2015; Ray et al. 2006] global integration with no correction; (ii) PGP-based global integration with local curl correction [Ray et al. 2006]; (iii) global integration with mixed-integer programming [Bommes et al. 2009]; and (iv) mixed-integer parameterization using an anisotropic metric (we used anisotropy factor 0.001). We show our result on the third row. Streamline of the vector fields are shown in blue, while the recovered isolines are shown in purple. We shade the models by the angle of deviation between the tangents to the integral curves, and the input vector field. (Black means aligned.) Our method simultaneously offers the usual benefits of PGP-based integration—integral curves well-aligned to the vector fields, and maintaining nice spacing between curves—while being robust to singularities.

singularities. For comparison, we also show ablation experiments where we compute a parameterization with no rescaling of the vector field (skipping step 3); and when using local curl reduction [Ray et al. 2006] to initialize  $s$  rather than our step 3.

*Comparison to Mixed-Integer Approaches.* In principle, the periodic global integration of Section 4.2.4 could be replaced by a mixed-integer approach, as in Bommes et al. [2009]’s mixed-integer quadrangulation algorithm. We compare against this alternative in Figure 15. In its raw form, the MIQ algorithm does not force exact alignment of  $\nabla\theta$  to the field (which is beneficial when quad meshing, to avoid anisotropic stretching of the quadrangulation streamlines), and we observe that the method yields isolines that do not align as well to the input field.

To promote alignment with the input field, several recent papers [Bommes et al. 2009; Campen et al. 2016] suggest adopting an

anisotropic metric when computing  $\theta$ :

$$\arg \min_{\theta} \sum_{i \in \mathcal{F}} \|[\nabla\theta]_i - \mathbf{w}_i^\perp\|_{g_i}^2, \quad g_i = \mathbf{w}_i \otimes \mathbf{w}_i + \alpha \mathbf{w}_i^\perp \otimes \mathbf{w}_i^\perp,$$

where  $\alpha = 1$  is the standard inner product on vector fields, and  $\alpha = 0$  penalizes only components of  $\nabla\theta$  orthogonal to  $\mathbf{w}^\perp$ . In Figure 15 we evaluate this variant of global integration as well. Although an anisotropic metric promotes alignment to the input field, this comes at the cost of very poorly-spaced integral curves for examples with complex geometry and topology; there unfortunately is no free lunch when choosing  $\alpha$ . As has been observed by others comparing these techniques [Bommes et al. 2013; Knöppel et al. 2015], global integration using PGP (with our  $s$ -initialization step) benefits from the ability to introduce edge dislocations in regions where integral curves diverge, yielding overall higher-quality foliations.

## 5 PART 2: DESIGNING TRIAXIAL WEAVES

We now apply our geodesic foliation optimization tools to the inverse triaxial weaving problem: given an embedded surface  $M$  in  $\mathbb{R}^3$ , how do we weave it out of flexible ribbons?

Ribbons in a triaxial weave can be grouped (away from singularities) into one of three nearly-parallel *families*, with ribbons of different families crossing at roughly sixty degrees to each other. As mentioned in the introduction, thin flexible ribbons resist geodesic curvature, and must follow approximate geodesics along the target surface  $M$ . In this section we create a pipeline for weaving arbitrary shapes out of ribbons.

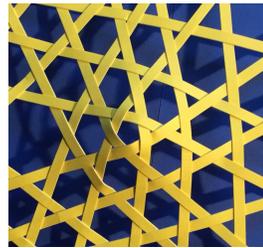
Since the three ribbon families of a triaxially-woven object lie along geodesics, a natural idea is to lay out ribbons along isolines of three independent geodesics foliations  $\{\theta_i\}_{i=1,2,3}$  of  $M$  (one for each family). However, designing each family independently does not allow for creation of *dislocation singularities* in the weave pattern (see inset). We will see that by instead solving for a *single* geodesic foliation on a sixfold branched cover of  $M$ , we can exploit singularities to improve the weave design.

More specifically, the summary of our strategy, described in detail in the remainder of this section, is the following: we place branch points and branch cuts on the input surface  $\mathcal{T}$  using the approach detailed in Appendix D. We then cut and glue six copies of  $\mathcal{T}$  into a sixfold cover  $C$  of  $\mathcal{T}$ : this cover is a manifold triangle mesh with boundary (though it may have multiple connected components, and may not be simply-connected even if the original surface is), and so the pipeline from Part 1 can be applied directly to  $C$  to find a geodesic foliation  $\theta : C \rightarrow S^1$  on the cover. After computing this foliation on the cover, we trace isolines of  $\theta$  on  $C$  and realize a weaving pattern by projecting these curves back onto the original mesh  $\mathcal{T}$ .

Our heuristic for placing branch points encourages approximate 3-fold symmetry of the final weaving pattern.

*Weave Dislocations.* Although ribbons in a triaxial weave can be grouped *locally* into three families, singularities are apparent in real-world weaves of nontrivially curved objects, and such singularities are neither avoidable (due to the Hopf index theorem) nor undesirable, as singularities are a powerful tool for designing weaves with curvature.

Singularities in the weave can be of two types: ordinary geometric singularities, where ribbons of one family converge to a point, as seen in the geodesic fields in Figure 8, and *topological* singularities where ribbons of a single family split around the singularity into two different families, as illustrated in figure 16.



Dislocations in a weave [Martin 2018] induce cone singularities where curvature concentrates. Notice that away from the singularity, ribbons can be grouped into three nearly-parallel families; one circulation around the singularity permutes the identity of the three ribbon families (the singularity has index  $\frac{2}{3}$ .)

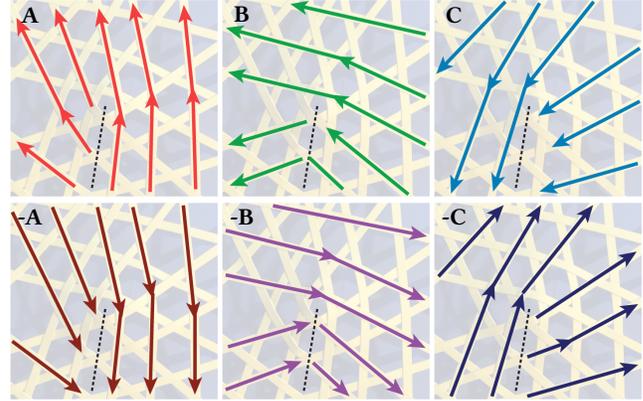


Fig. 16. Circulating around topological singularities (here, a simplified and labeled reduction of Martin’s yellow ribbon weave) alters the identity and orientation of ribbon families: in this case, moving counterclockwise around the singularity (of index  $\frac{1}{3}$ ) permutes the family labeling cyclically by  $(A, B, C, -A, -B, -C) \in S_6$ .

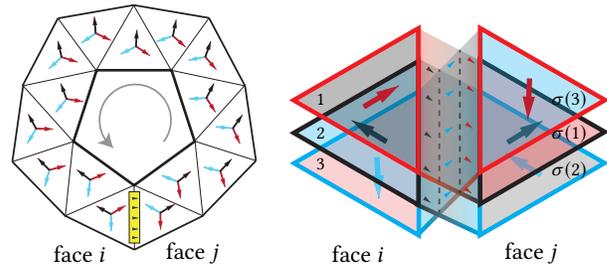


Fig. 17. We represent triaxial weaves of a surface as geodesic foliations on a six-fold cover of that surface, with branch points of the cover corresponding to weave dislocations. We initialize the branch points of a 6-RoSy field: vertices with fractional holonomy ( $\frac{1}{3}$  for the figure on *left*) become branch points. We first remove the 1-ring of all singular vertices, then split the resulting mesh into 6 copies. We re-glue each copy of a face  $i$  to a copy of a neighboring face  $j$  based on the permutation  $\sigma_{ij}$  which optimally assigns vectors of the RoSy on face  $i$  to those on face  $j$  (*right*).

Following ideas from cross field parameterization [Kälberer et al. 2007] we approach the problem of globally integrating a poly-vector field with topological singularities by lifting to a branched cover, and illustrate our discretization in figure 17. However, borrowing an observation from the study of one-directional stripe patterns on surfaces [Knöppel et al. 2015], we note that in a weave, ribbons are oriented, and so ribbons in the same family may actually travel anti-parallel to each other. We thus represent a triaxial weave as a *single* approximately-geodesic foliation  $\theta$  on a sixfold branched cover  $C$  of  $\mathcal{T}$ . Each face of  $\mathcal{T}$  lifts to *six* faces, one on each cover, and likewise isolines of  $\theta$  project down to three families of curves on  $\mathcal{T}$ .

### 5.1 Branch Point Initialization

Constructing the branched cover  $C$  requires choosing the branch points and branch cuts on the original surface. We do not attempt

to solve the interesting, yet challenging, problem of jointly optimizing for the cover topology and geodesic foliation, opting instead to first fix the topology of  $C$  and then solve for  $\theta$  on that fixed covering space. Our heuristic for fixing the topology is based on the observation that a globally-optimal [Knöppel et al. 2013] (as-smooth-as-possible) 6-RoSy field on a surface distributes singularities over the surface in a manner that minimizes distortion of the RoSy field. While not optimal, we find that this approach is generally suitable for optimizing geodesic foliations for fabrication. We did not need to hand-tune singularity placement for any of the examples in this paper, including the examples in Figure 24.

Specifically, after computing a globally optimal 6-RoSy on the faces of  $\mathcal{T}$ , we split the RoSy into six vectors per face, and for each edge  $e_{ij} \in \mathcal{E}$  we compute an optimal permutation  $\sigma_{ij} \in S_6$  assigning each vector on face  $i$  to one on face  $j$ , so as to minimize the sum of squared angles between corresponding vectors. Although in principle the entire foliation-optimization pipeline of Part 1 could be implemented on an abstract covering space represented by these permutations [Kälberer et al. 2007], we found that explicitly materializing  $C$  as a triangle mesh using a gluing construction similar to the one described by Roy et al. [2018] was conceptually simple and more straightforward to implement.

In our implementation, we first detect branch points on  $\mathcal{T}$ , i.e. vertices around which the permutations  $\sigma$  do not compose to the identity. We puncture  $\mathcal{T}$  by removing these branch points and all neighboring faces. Then we create six copies of each remaining face and glue copy  $k$  of face  $i$  to copy  $\sigma_{ij}(k)$  of neighboring face  $j$ , as illustrated in Figure 17. Note that while the resulting surface  $C$  is a manifold, it may contain multiple connected components. Finally, we also remove geometric singularities from  $C$  by similarly puncturing a neighborhood of each singularity, to allow for recovery of a continuous  $\theta$  on the branched cover.

## 5.2 Pipeline Overview

Figure 18 illustrates our pipeline for computing a triaxial weave approximating a given surface  $\mathcal{T}$  (in this case, a 3D scan of brain aneurysm). The main steps of our pipeline are:

- **Cover Initialization.** We intrinsic-Delaunay-remesh  $\mathcal{T}$  and create a branched cover  $C$  of  $\mathcal{T}$  using the RoSy heuristic described in the previous section. The RoSy lifts to a plain unit vector field  $\hat{w}^0$  on  $C$ .
- **Geodesic Foliation Optimization.** Using  $\hat{w}^0$  as an initial guess, we apply the pipeline from Section 4 to compute an approximately-geodesic almost-foliation  $\theta$  on every connected component of the cover  $C$ .
- **Paired Cover Rounding.** Each face of  $\mathcal{T}$  lifts to six faces of  $C$ , and these six faces can be grouped into three pairs on which the isolines of  $\theta$  correspond to ribbons with identical direction, but opposite orientation. To ensure that isolines on paired covering faces are well-spaced relative to each other, we round  $\theta$  to differ by exactly half of a phase shift on paired faces, using mixed-integer programming and the CoMiSo library [2012] as described by, for instance, Kahlbehrer et al. [2007].
- **Centerline Extraction.** We extract isolines of  $\theta$  on the six-fold cover. These isolines approximate the centerlines of ribbons in

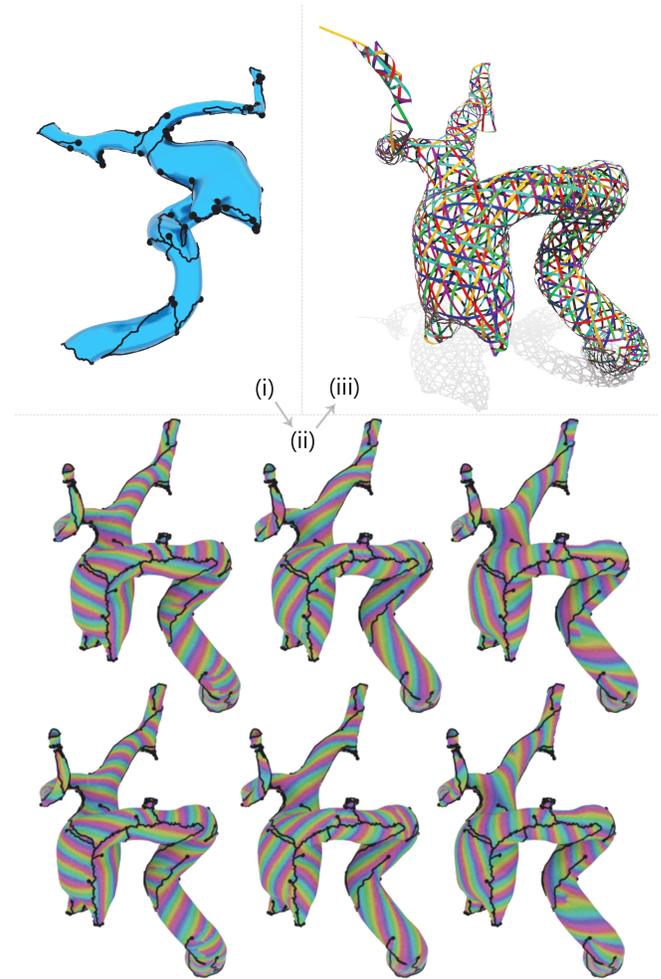


Fig. 18. Our weave design pipeline, illustrated on a 3D reconstruction of a brain aneurysm. We solve for a globally optimal 6-RoSy field [Knöppel et al. 2013] to place singularities on the model (i). We then optimize an approximately geodesic vector (Section 4.1) on that cover. From there, we lift to a branched cover and solve for periodic scalar valued function which aligns with the geodesic vector field (Section 4.2), (ii). The level sets of this function specify the ribbon centerlines (see Figure 4). We trace these and project them from  $C$  back to  $\mathcal{T}$  to materialize a weaving pattern, and then fine tune the pattern via a physics simulation (iii).

a weaving pattern. However, due to numerical aliasing, isolines which pass close to edge dislocations (like those discussed in Figure 11, right) may be highly non-geodesic at points. Further, our technique of removing faces in the neighborhood of topological singularities can also result in disconnected weaves that come undone during simulation (see Figure 22, right). To account for these artifacts, we post-process the isolines in several ways. We:

- resample the isolines so that all segments are approximately the same length;
- cut the ribbons into two pieces at points with very high amounts of geodesic curvature;

- prune very short ribbon short segments;
- ensure that ribbons terminate at crossings without “loose ends,” by slightly extending ribbons along geodesics and then subsequently trimming them to the nearest crossing. The process also helps “patch” the weave near singularities.
- **Crossing Assignment.** We detect crossings of ribbons and perform over/under assignment. Ideally, each ribbon should alternate going over and under the ribbons it intersects; finding a global over-under assignment can be viewed as a graph-coloring problem, where the vertices are contact points, edges connect pairs of contact points that are consecutive on some ribbon, and we seek a two-coloring of the vertices (representing an over or under crossing) that minimizes monochromatic edges. We use a greedy heuristic to approximately solve this NP-hard problem [Poljak and Tuza 1994]: we choose a ribbon at random, sort its crossings, and assign it as the over/under ribbon in alternation. We repeat this process for all other ribbons. In the case where a ribbon’s desired assignment is impossible (because another ribbon has already forced it in the opposite position) we respect the prior decision and resume alternation.
- **Physics Fine-tuning.** We improve the quality of the weave by running an elastic rod simulation of the ribbons, allowing ribbons to slide tangent to the surface  $M$  to release elastic strain. We then predict the final equilibrium shape of the weave under its own weight, to validate the weave design before fabrication. Details of this simulation and fine-tuning step are given in Appendix E. During this simulation, we allow ribbons to rotate relative to each other, which produces a concrete set of joint angles which we leverage for fabrication.
- **Fabrication.** We export the weave pattern as a set of flat ribbons annotated with crossing position and angle information; the pattern can then be printed out and physically assembled.

### 5.3 Pipeline Results

The gallery in Figure 24 shows triaxial weaves computed on a variety of different surfaces. Our method successfully generates stable weaving patterns that closely approximate input models with a wide variety of geometries and topologies. A reference implementation of our algorithms and design pipeline is included in the supplemental materials; we used this code to generate all examples in the paper. As evidence of the robustness of our pipeline, all of these examples were generated in a single click in our software package with no parameter tuning. (We did tune the global scaling of  $\theta$  in Figures 1 and 19 to ease fabrication burden by designing patterns with fewer ribbons.)

### 5.4 Fabrication

We develop each ribbon into the plane as parallel strips of width  $w$ , and we export the ribbons as a color-coded and labeled vector graphics file, illustrated in Figure 20. Our assembly language is similar to the one used by the modeling program WeaveMesh [Tao et al. 2017], with one key difference: in addition to marking on each color-coded and labeled ribbon the locations of its crossings, we also draw guides on the ribbons showing the in-plane angles at which they cross, which we find greatly aids in the incremental assembly

of geometrically involved forms. The angle information allows the weaver to handle each crossing only once, with the confidence that the crossing orientation will not need to be readjusted later.

A fabrication strategy we have found to be effective is to begin by assembling several medium length ribbons to serve as the “backbone” of the woven structure, and to build everything else off of this backbone. Even so, our more elaborate examples require substantial manual effort to fabricate (about fifty hours for the bunny in Figure 1), owing to the sheer number of ribbons in the weave, the time required to find the correct position and orientation of each rod at each crossing, and to correct mistakes—even a few errors early on may lead to significant distortions in the assembled form.

The fabricated examples in Figure 19 were woven from Bristol vellum, as vellum allows us to print the ribbon and guide markings directly onto a structural weaving material. The bunny in Figure 1 was fabricated from birch edge-banding. This material is cheap, widely available (with adhesive conveniently pre-applied to one side), and structurally more robust than vellum. For this example we printed the design file onto transfer paper, which we affixed to the edge-banding prior to assembly. This model consisted of 108 ribbons and 1823 ribbon crossings. It stands approximately 1.5 meters tall, and required 180m of edge banding to fabricate. In contrast, the torus in Figure 19 consists of 10 ribbons and 84 crossings.

Though quite work intensive, our fabrication methodology allows non-experts to fabricate large scale spacial structures which would be difficult to realize any other way. We encourage the reader to try the system for themselves by fabricating a woven cone; to do so, print out and assemble the weave pattern in appendix F. You will need to fasten the ribbons together at their crossings (using glue, tape, or binder clips).

## 6 CONCLUSIONS, LIMITATIONS, AND FUTURE WORK

We have presented new algorithms for designing geodesic foliations on triangulated surfaces, and we have demonstrated how these geometry processing tools can be applied to the problem of weaving arbitrary curved surfaces. We have validated our techniques on a variety of complex examples. There are, however, several limitations of our pipeline that could be explored in future work, and a number of avenues for more exploration.

*Tessellation-Independence.* Each variational problem in Section 4 has been designed to have solutions which are invariant to the scale of  $\mathcal{T}$  and resilient to its tessellation. That said, we make no claims of guaranteed tessellation-independence, and indeed our formulation requires norms on face-based vector fields, the Laplacian operator on functions on faces, etc, whose discretization remains an active area of research [Alexa and Wardetzky 2011]. That said, quality of the input tessellation is not a practical concern since there is no reason not to remesh all inputs to have high quality while keeping the shape the same.

*Weave Fidelity.* It is clear that there is a limit on the scale of geometric features that can be resolved by a weave pattern, depending on both the ribbon width and density. Sharp creases are also unlikely to be preserved (for instance, the icosahedron in the first column of Figure 19, despite possessing a perfectly regular 6-RoS)

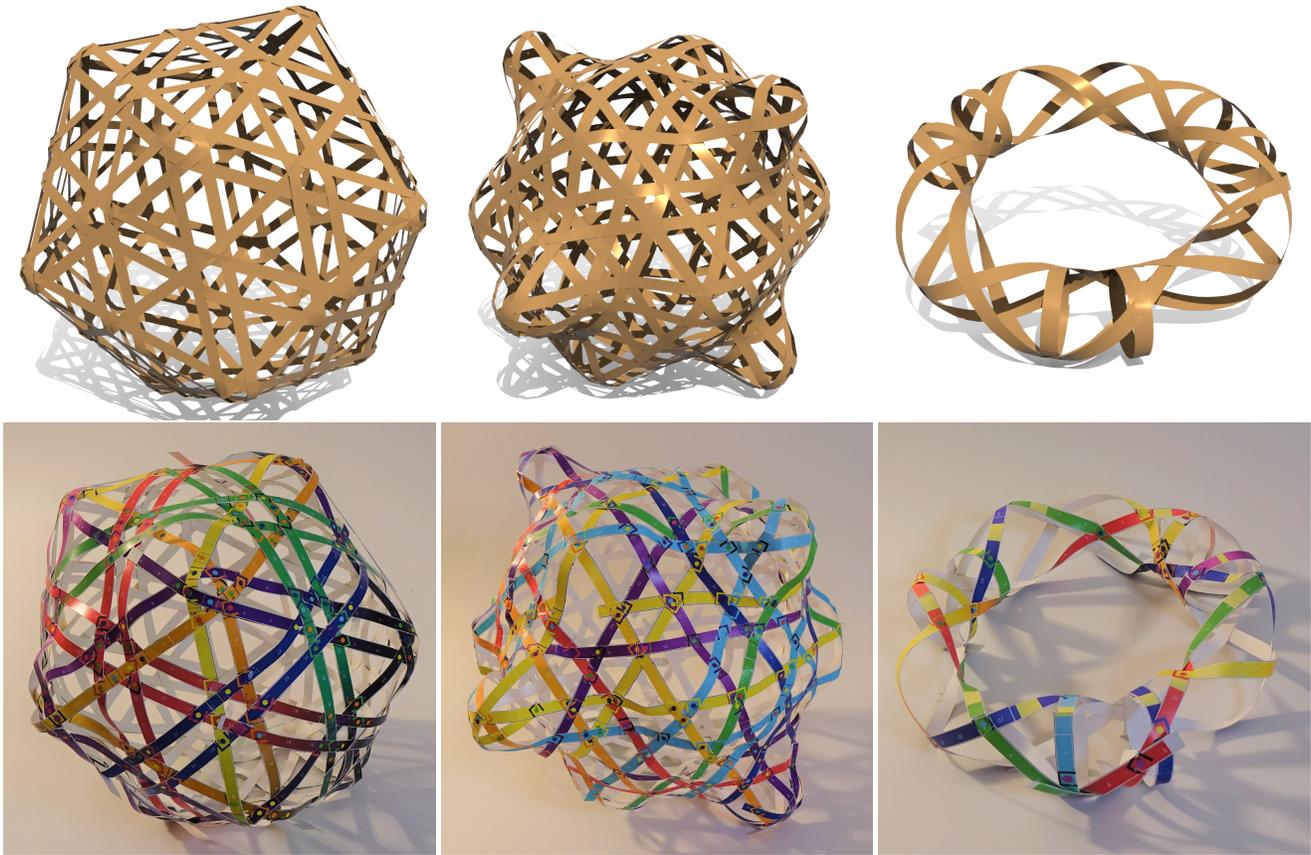


Fig. 19. The results of our weave design pipeline (top), and the corresponding fabricated model, for an icosahedron, bumpy box, and torus.

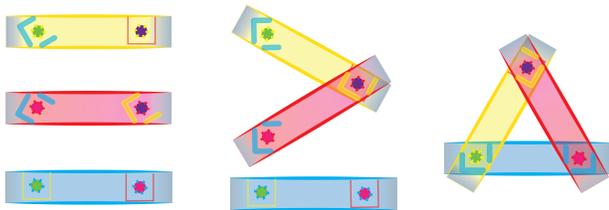


Fig. 20. Our visual assembly language—we mark each intersection with an angle and an indicator pattern to match ribbons to each other. The indicators are drawn from a small set of visually distinct crossings/patterns. We find that these greatly aid in accelerating assembly, but also uniquely label each crossing for disambiguation.

geodesic foliation, loses its sharp creases when the ribbons relax to static equilibrium.) Higher-fidelity weaves require more and thinner ribbons.

*Crossing Assignment and Rigidity.* Triaxial structures tend to be truss-like and rigid, particularly if the crossing assignment is chosen so that three ribbons from the three ribbon families intersecting

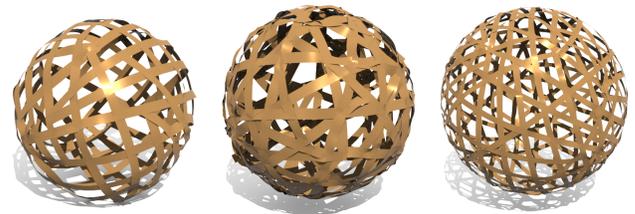


Fig. 21. Auto-Chinlone: our pipeline easily extends to weaves with arbitrary  $k$ -fold symmetry, and we illustrate the different choices of  $k$  on the sphere (from left to right:  $k = 4, 5, 6$ ).

each other in close proximity interleave (see Figure 20, right). Currently, our crossing assignment heuristic is ad-hoc and does not take advantage of this observation. Better automatic crossing assignment selection would reduce the need to fasten or manually correct crossings and increase rigidity during fabrication. However, solving this problem in generality requires accounting for ribbon tolerances (they have width and cannot pass through each other) in addition to regularity when assigning the ribbon pattern. More sophisticated

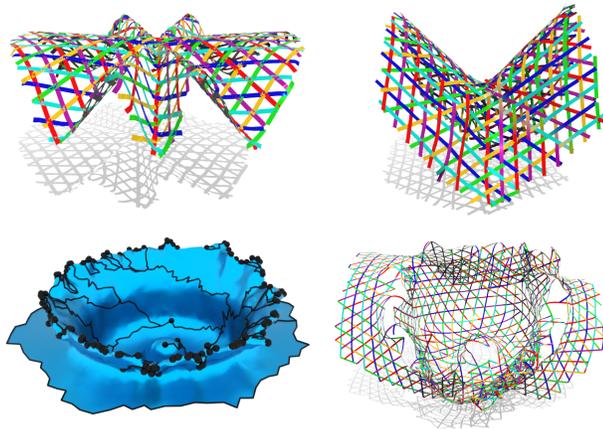


Fig. 22. Some models, particularly those with large amounts of negative curvature and with boundary, may relax to equilibrium (*top right*) far away from their initial shape (*top left*). Our singularity placement heuristic can also result in an unreasonable number of singularities, disconnecting seams on some models and leading to significant distortion, when the input surface contains sharp, noisy features, such as a crowning water droplet (*bottom*)

approximations to the Max-Cut crossing assignment problem [Goe-mans and Williamson 1995], with theoretical guarantees, also exist and would be interesting to explore.

Further, while we find that the space of stable woven geometries is extremely rich, in some cases (particularly for negatively-curved surfaces with free boundary), the weave design relaxes to an equilibrium shape far from the input shape after simulation (see Figure 22 for an example). Developing a more complete understanding of the space of stable surfaces (and how they respond to external loads) is an interesting direction for future work.

*Singularity Placement.* When designing weaves, we automatically places branch cuts based on the singularities of an as-smooth-as-possible 6-RoSy field [Knöppel et al. 2013]. While this heuristic requires no user intervention and gives good results in practice, we have no theoretical guarantees that this choice of singularity placement is optimal (and indeed, more singularities might allow a more geodesic field, and vice versa). We noticed that in spite of the rigidity induced by fixing a branched cover prior to optimization, our geodesic optimization finds weave patterns that are close to geodesic and have low elastic strain (see Figure 23). There is interesting future work in combining our approach with automatic singularity placement; one natural approach would be to extend our geodesic foliation formulation to use PolyVectors [Diamanti et al. 2014], so that the geometry and topology of the branched cover does not need to be fixed a priori. Another option towards a similar goal would be to follow the approach of Farchi and Ben-Chen [2018] and to use our energy as a metric to power a search over the space of singularities.

*Ribbon Degeneracy.* In order to be fabricable, ribbon families should cross at angles close to sixty degrees. Our current pipeline does not guarantee that the isolines of  $\theta$  on the cover  $C$  project to curves on  $\mathcal{T}$  that meet transversely, and including additional terms

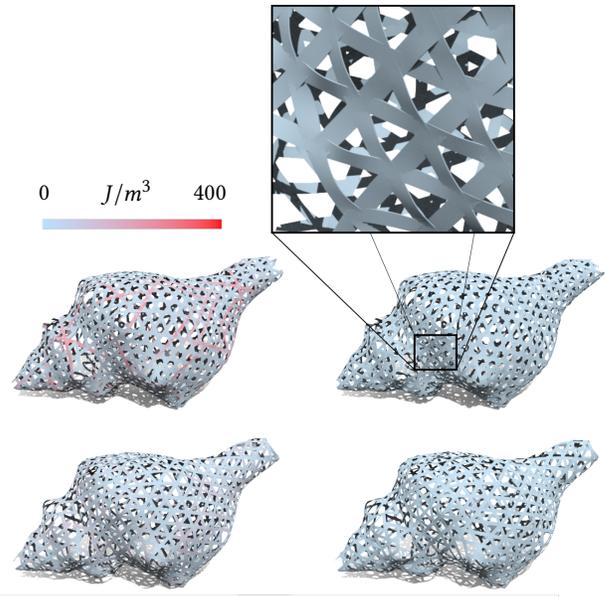


Fig. 23. *Left:* a weave aligned to a non-geodesic foliation (a smooth 6-RoSy field [Knöppel et al. 2013], top) and a geodesic foliation that we compute (bottom). Curvature of the ribbon centerlines induces elastic energy in the ribbons, with geodesic curvature contributing significantly more energy; we plot the energy density as a color map on the ribbons. *Right:* Our simulation relaxes the weaves to equilibrium. The inset shows the typical buckling and twisting that occurs if non-geodesics are forced into an unrelaxed weave pattern.

in the optimization of  $\theta$  to encourage ribbons to meet at robust angles might improve the quality of our weaves.

*Other Weave Types.* Our work focuses on triaxial weaves, as they are the most geometrically and topologically complex; in principle our theory and fabrication pipeline could be extended in a straightforward manner to design of biaxial (plain, twill) weaves. Such structures are likely to be less rigid and so might be useful for the design of forms which respond to external forces in pre-determined ways. As we demonstrate in Figure 21, our pipeline readily extends to weaving patterns with arbitrary degrees of symmetry.

*Alternative Vector Field Discretization.* We discretize vector fields as assignments of tangent vectors to mesh faces: the face defines a natural and obvious tangent plane in which the vectors lie, it allows us to build on prior work tying together discrete curl and integrability of face-based vectors, and it is a simple representation to implement. Porting the geodesic foliation algorithms in Section 4 to other discretizations [de Goes et al. 2016] (such as vectors on vertices or edges, or higher-order bases for vector fields) may be possible as well.

*Automated Fabrication.* All of our real-world models were fabricated by hand, with substantial manual effort. Research into automating fabrication of weaves from our optimized weave patterns,

along the lines of what has been done for knitted 3D objects [McCann et al. 2016; Narayanan et al. 2018], is an important next step towards bulk manufacturing of complex woven objects.

## ACKNOWLEDGMENTS

We'd like to thank the anonymous reviewers for their insightful feedback and encouragement over several revisions. Ridwan Syed for pointing out the connection between plain weaving and Max-Cut. Kevin Sun for contributing to the initial physics simulator implementation. Rawan Al-Qarqaz, Sepideh Maleki, Kelsey Willard, Matt Jordan, Nereida Lewis-Ramirez, Anna Yesypenko, and a host of others who helped assemble various woven prototypes. This work was sponsored in part by gifts from Adobe Research and SideFX software.

## REFERENCES

- Hillel Aharoni, Desislava V. Todorova, Octavio Albarrán, Lucas Goehring, Randall D. Kamien, and Eleni Katifori. 2017. The smectic order of wrinkles. In *Nature communications*.
- Ergun Akleman, Jianer Chen, YenLin Chen, Qing Xing, and Jonathan L. Gross. 2011. Cyclic twill-woven objects. *Computers & Graphics* 35, 3 (2011), 623–631. <https://doi.org/10.1016/j.cag.2011.03.003> Shape Modeling International (SMI) Conference 2011.
- Ergun Akleman, Jianer Chen, Qing Xing, and Jonathan L. Gross. 2009. Cyclic Plain-weaving on Polygonal Mesh Surfaces with Graph Rotation Systems. In *ACM SIGGRAPH 2009 Papers (SIGGRAPH '09)*. ACM, New York, NY, USA, Article 78, 8 pages. <https://doi.org/10.1145/1576246.1531384>
- Marc Alexa and Max Wardetzky. 2011. Discrete Laplacians on General Polygonal Meshes. *ACM Trans. Graph.* 30, 4, Article 102 (July 2011), 10 pages. <https://doi.org/10.1145/2010324.1964997>
- Patricio Aviles and Yoshikazu Giga. 1987. A mathematical problem related to the physical theory of liquid crystal configurations. In *Miniconference on geometry/partial differential equations*, 2. Centre for Mathematical Analysis, The Australian National University, Canberra AUS, 1–16. <https://projecteuclid.org/euclid.pcma/1416336633>
- Phil Ayres, Alison Grace Martin, and Mateusz Zwierzycki. 2018. Beyond the basket case: A principled approach to the modelling of kagome weave patterns for the fabrication of interlaced lattice structures using straight strips. In *Advances in Architectural Geometry*.
- Omri Azenkot, Maks Ovsjanikov, Frédéric Chazal, and Mirela Ben-Chen. 2015. Discrete Derivatives of Vector Fields on Surfaces – An Operator Approach. *ACM Trans. Graph.* 34, 3, Article 29 (May 2015), 13 pages. <https://doi.org/10.1145/2723158>
- Mirela Ben-Chen, Adrian Butscher, Justin Solomon, and Leonidas Guibas. 2010. On Discrete Killing Vector Fields and Patterns on Surfaces. *Computer Graphics Forum* 29, 5 (2010), 1701–1711. <https://doi.org/10.1111/j.1467-8659.2010.010779.x> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2010.010779.x>
- Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2010. Discrete Viscous Threads. *ACM Trans. Graph.* 29, 4, Article 116 (July 2010), 10 pages. <https://doi.org/10.1145/1778765.1778853>
- Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete Elastic Rods. *ACM Trans. Graph.* 27, 3, Article 63 (Aug. 2008), 12 pages. <https://doi.org/10.1145/1360612.1360662>
- Florence Bertails, Basile Audoly, Marie-Paule Cani, Bernard Querleux, Frédéric Leroy, and Jean-Luc Lévêque. 2006. Super-helices for Predicting the Dynamics of Natural Hair. In *ACM SIGGRAPH 2006 Papers (SIGGRAPH '06)*. ACM, New York, NY, USA, 1180–1187. <https://doi.org/10.1145/1179352.1142012>
- H. Bhatia, S. Jadhav, P. Bremer, G. Chen, J. A. Levine, L. G. Nonato, and V. Pascucci. 2011. Edge maps: Representing flow with bounded error. In *2011 IEEE Pacific Visualization Symposium*. IEEE, Hong Kong, China, 75–82. <https://doi.org/10.1109/PACIFICVIS.2011.5742375>
- E. Boeckx and L. Vanhecke. 2000. Harmonic and minimal vector fields on tangent and unit tangent bundles. *Differential Geometry and its Applications* 13 (07 2000), 77–93. [https://doi.org/10.1016/S0926-2245\(00\)00021-8](https://doi.org/10.1016/S0926-2245(00)00021-8)
- David Bommes, Bruno Lévy, Nico Pietroni, Enrico Puppo, Claudio Silva, Marco Tarini, and Denis Zorin. 2013. Quad-Mesh Generation and Processing: A Survey. *Comput. Graph. Forum* 32, 6 (Sept. 2013), 51–76. <https://doi.org/10.1111/cgf.12014>
- David Bommes, Henrik Zimmer, and Leif Kobbelt. 2009. Mixed-integer Quadrangulation. *ACM Trans. Graph.* 28, 3, Article 77 (July 2009), 10 pages. <https://doi.org/10.1145/1531326.1531383>
- David Bommes, Henrik Zimmer, and Leif Kobbelt. 2012. Practical Mixed-integer Optimization for Geometry Processing. In *Proceedings of the 7th International Conference on Curves and Surfaces*. Springer-Verlag, Berlin, Heidelberg, 193–206. [https://doi.org/10.1007/978-3-642-27413-8\\_12](https://doi.org/10.1007/978-3-642-27413-8_12)
- Boston Scientific. 2015. Innova: Self-expanding nitinol stent system for Superficial Femoral Artery. <http://www.bostonscientific.com/en-EU/products/stents-vascular/innova-self-expanding-stent-system.html>. Accessed: 2018-05-27.
- J. B. Boyling. 1968. Carathéodory's principle and the existence of global integrating factors. *Communications in Mathematical Physics* 10, 1 (01 Feb 1968), 52–68. <https://doi.org/10.1007/BF01654133>
- P.-T. Brun, Basile Audoly, Neil M. Ribe, T. S. Eaves, and John R. Lister. 2015. Liquid Ropes: A Geometrical Model for Thin Viscous Jet Instabilities. *Phys. Rev. Lett.* 114 (Apr 2015), 174501. Issue 17. <https://doi.org/10.1103/PhysRevLett.114.174501>
- Marcel Campen, David Bommes, and Leif Kobbelt. 2012. Dual Loops Meshing: Quality Quad Layouts on Manifolds. *ACM Trans. Graph.* 31, 4, Article 110 (July 2012), 11 pages. <https://doi.org/10.1145/2185520.2185606>
- Marcel Campen, Moritz Ibing, Hans-Christian Ebke, Denis Zorin, and Leif Kobbelt. 2016. Scale-Invariant Directional Alignment of Surface Parametrizations. *Computer Graphics Forum* 35, 5 (2016), 1–10. <https://doi.org/10.1111/cgf.12958> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12958>
- Marcel Campen and Leif Kobbelt. 2014a. Dual Strip Weaving: Interactive Design of Quad Layouts Using Elastica Strips. *ACM Trans. Graph.* 33, 6, Article 183 (Nov. 2014), 10 pages. <https://doi.org/10.1145/2661229.2661236>
- M. Campen and L. Kobbelt. 2014b. Quad Layout Embedding via Aligned Parameterization. *Comput. Graph. Forum* 33, 8 (Dec. 2014), 69–81. <https://doi.org/10.1111/cgf.12401>
- Paolo Cignoni, Nico Pietroni, Luigi Malomo, and Roberto Scopigno. 2014. Field-aligned Mesh Joinery. *ACM Trans. Graph.* 33, 1, Article 11 (Feb. 2014), 12 pages. <https://doi.org/10.1145/2537852>
- Gabriel Cirio, Jorge Lopez-Moreno, David Miraut, and Miguel A. Otaduy. 2014. Yarn-level Simulation of Woven Cloth. *ACM Trans. Graph.* 33, 6, Article 207 (Nov. 2014), 11 pages. <https://doi.org/10.1145/2661229.2661279>
- Keenan Crane, Mathieu Desbrun, and Peter Schröder. 2010. Trivial Connections on Discrete Surfaces. *Computer Graphics Forum* 29, 5 (2010), 1525–1533. <https://doi.org/10.1111/j.1467-8659.2010.01761.x> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2010.01761.x>
- Keenan Crane, Clarisse Weischedel, and Max Wardetzky. 2013. Geodesics in Heat: A New Approach to Computing Distance Based on Heat Flow. *ACM Trans. Graph.* 32, 5, Article 152 (Oct. 2013), 11 pages. <https://doi.org/10.1145/2516971.2516977>
- Fernando de Goes, Mathieu Desbrun, and Yiying Tong. 2016. Vector Field Processing on Triangle Meshes. In *ACM SIGGRAPH 2016 Courses (SIGGRAPH '16)*. ACM, New York, NY, USA, Article 27, 49 pages. <https://doi.org/10.1145/2897826.2927303>
- Olga Diamanti, Amir Vaxman, Daniele Panozzo, and Olga Sorkine-Hornung. 2014. Designing N-PolyVector Fields with Complex Polynomials. *Comput. Graph. Forum* 33, 5 (Aug. 2014), 1–11. <https://doi.org/10.1111/cgf.12426>
- Olga Diamanti, Amir Vaxman, Daniele Panozzo, and Olga Sorkine-Hornung. 2015. Integrable PolyVector Fields. *ACM Trans. Graph.* 34, 4, Article 38 (July 2015), 12 pages. <https://doi.org/10.1145/2766906>
- N. M. Ercolani and S. C. Venkataramani. 2009. A Variational Theory for Point Defects in Patterns. *Journal of Nonlinear Science* 19, 3 (01 Jun 2009), 267–300. <https://doi.org/10.1007/s00332-008-9035-9>
- Nahum Farchi and Mirela Ben-Chen. 2018. Integer-only Cross Field Computation. *ACM Trans. Graph.* 37, 4, Article 91 (July 2018), 13 pages. <https://doi.org/10.1145/3197517.3201375>
- Matthew Fisher, Peter Schröder, Mathieu Desbrun, and Hugues Hoppe. 2007. Design of Tangent Vector Fields. *ACM Trans. Graph.* 26, 3 (July 2007), 1–9. <https://doi.org/10.1145/1276377.1276447>
- Marc Fornes. 2017. Minima | Maxima. World Expo 2017.
- Akash Garg, Andrew O. Sageman-Furnas, Bailin Deng, Yonghao Yue, Eitan Grinspun, Mark Pauly, and Max Wardetzky. 2014. Wire Mesh Design. *ACM Trans. Graph.* 33, 4, Article 66 (July 2014), 12 pages. <https://doi.org/10.1145/2601097.2601106>
- Michel X. Goemans and David P. Williamson. 1995. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *J. ACM* 42, 6 (Nov. 1995), 1115–1145. <https://doi.org/10.1145/227683.227684>
- Aaron Hertzmann and Denis Zorin. 2000. Illustrating Smooth Surfaces. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 517–526. <https://doi.org/10.1145/344779.345074>
- W. S. Howard and V. Kumar. 1993. A minimum principle for the dynamic analysis of systems with frictional contacts. In *[1993] Proceedings IEEE International Conference on Robotics and Automation*. IEEE, Atlanta, GA, 437–442 vol.3. <https://doi.org/10.1109/ROBOT.1993.292211>
- Yuki Igarashi, Takeo Igarashi, and Hiromasa Suzuki. 2008. Knitting a 3D Model. *Comput. Graph. Forum* 27 (10 2008), 1737–1743. <https://doi.org/10.1111/j.1467-8659.2008.01318.x>
- Pierre-Emmanuel Jabin, Felix Otto, and Benoit Perthame. 2002. Line-energy Ginzburg-Landau models : zero-energy states. *Annali della Scuola Normale Superiore di Pisa - Classe di Scienze Ser. 5*, 1, 1 (2002), 187–202.

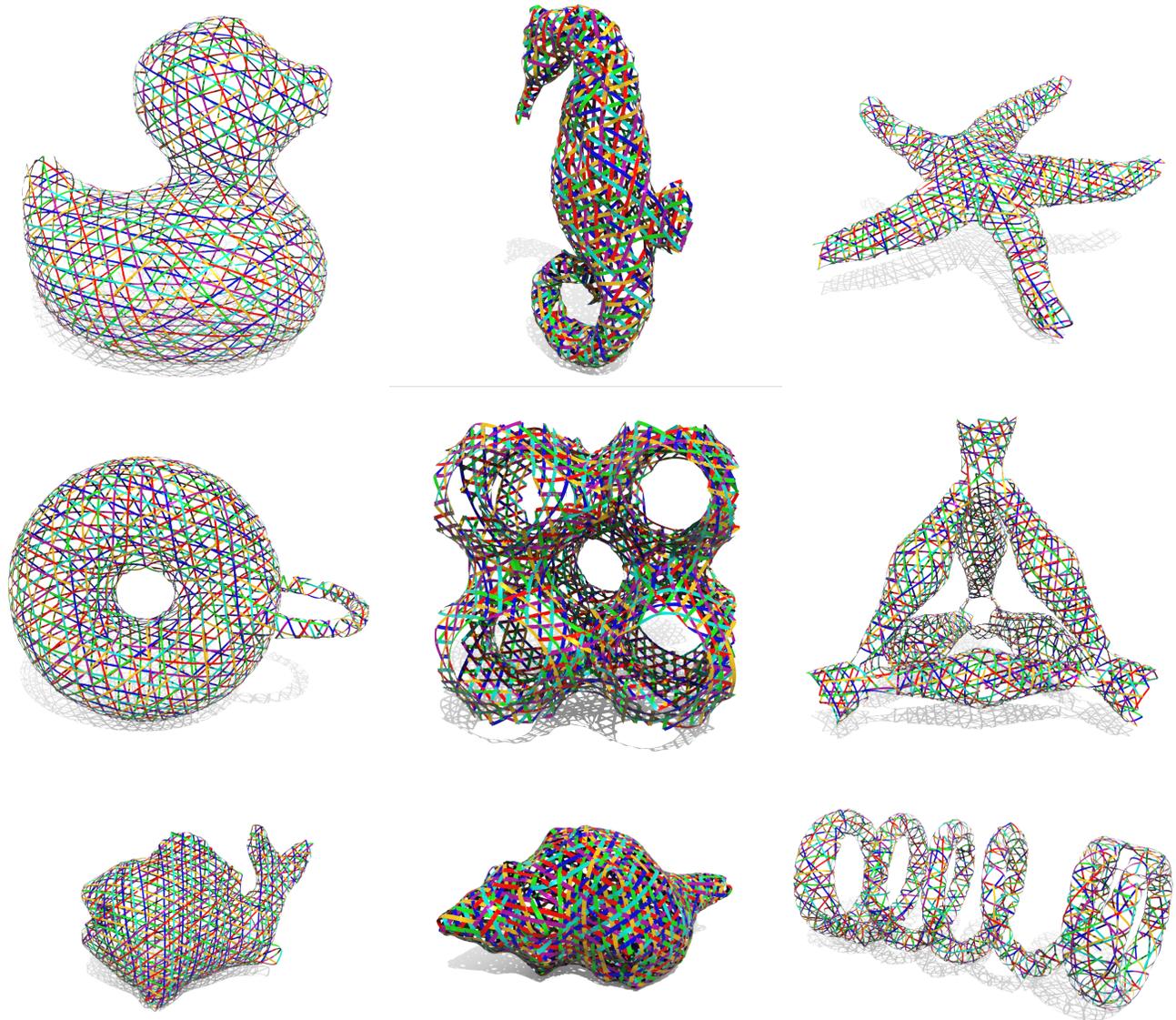


Fig. 24. Gallery of examples. All examples in this figure were generated by a fully automated pipeline with no manual placement of singularities or parameter-tuning (however physical fabrication does require some manual pruning of ribbons, adjustment of crossing assignment, etc.). The number of ribbons in each model, starting with the duck and the seahorse, are: 173, 123, 170, 177, 110, 658, 259, 115, 117, 283.

Wenzel Jakob, Marco Tarini, Daniele Panozzo, and Olga Sorkine-Hornung. 2015. Instant Field-aligned Meshes. *ACM Trans. Graph.* 34, 6, Article 189 (Oct. 2015), 15 pages. <https://doi.org/10.1145/2816795.2818078>

Felix Kälberer, Matthias Nieser, and Konrad Polthier. 2007. QuadCover – Surface Parameterization using Branched Coverings. *Comput. Graph. Forum* 26 (09 2007), 375–384.

Jonathan M. Kaldor, Doug L. James, and Steve Marschner. 2008. Simulating Knitted Cloth at the Yarn Level. In *ACM SIGGRAPH 2008 Papers (SIGGRAPH '08)*. ACM, New York, NY, USA, Article 65, 9 pages. <https://doi.org/10.1145/1399504.1360664>

W. Klingenberg. 1978. *Lectures on Closed Geodesics*. Springer-Verlag. <https://books.google.com/books?id=tInvAAAAMAAJ>

Felix Knöppel, Keenan Crane, Ulrich Pinkall, and Peter Schröder. 2013. Globally Optimal Direction Fields. *ACM Trans. Graph.* 32, 4, Article 59 (July 2013), 10 pages. <https://doi.org/10.1145/2461912.2462005>

Felix Knöppel, Keenan Crane, Ulrich Pinkall, and Peter Schröder. 2015. Stripe Patterns on Surfaces. *ACM Trans. Graph.* 34 (2015), 1–11. Issue 4.

Robert Kohn. 2006. Energy-driven pattern formation. In *International Congress of Mathematicians, ICM 2006*, Vol. 1. 359–383.

Jonathan Leaf, Rundong Wu, Eston Schweickart, Doug L. James, and Steve Marschner. 2018. Interactive Design of Periodic Yarn-level Cloth Patterns. *ACM Trans. Graph.* 37, 6, Article 202 (Dec. 2018), 202:1–202:15 pages. <https://doi.org/10.1145/3272127.3275105>

Urszula Lewandowska, Wojciech Zajaczkowski, Stefano Corra, Junki Tanabe, Ruediger Borrmann, Edmondo M. Benetti, Sebastian Stappert, Kohei Watanabe, Nellie A. K. Ochs, Robin Schaeublin, Chen Li, Eiji Yashima, Wojciech Pisula, Klaus MÄEllen, and Helma Wennemers. 2017. A triaxial supramolecular weave. *Nature Chemistry* 9 (24 Jul 2017), 1068 EP –. <http://dx.doi.org/10.1038/nchem.2823> Article.

Binbin Lin, Xiaofei He, Chiyuan Zhang, and Ming Ji. 2013. Parallel Vector Field Embedding. *Journal of Machine Learning Research* 14 (2013), 2945–2977.

- Beibei Liu, Yiyong Tong, Fernando De Goes, and Mathieu Desbrun. 2016. Discrete Connection and Covariant Derivative for Vector Field Analysis and Design. *ACM Trans. Graph.* 35, 3, Article 23 (March 2016), 17 pages. <https://doi.org/10.1145/2870629>
- Thomas Machon, Hillel Aharoni, Yichen Hu, and Randall D. Kamien. 2019. Aspects of Defect Topology in Smectic Liquid Crystals. *Communications in Mathematical Physics* (20 Feb 2019). <https://doi.org/10.1007/s00220-019-03366-y>
- Alison Martin. 2018. Alison Grace Martin. <http://gallery.bridgesmathart.org/exhibitions/2013-bridges-conference/alison-martin>. Accessed: 2018-01-23.
- James McCann, Lea Albaugh, Vidya Narayanan, April Grow, Wojciech Matusik, Jennifer Mankoff, and Jessica Hodgins. 2016. A Compiler for 3D Machine Knitting. *ACM Trans. Graph.* 35, 4, Article 49 (July 2016), 11 pages. <https://doi.org/10.1145/2897824.2925940>
- Eder Miguel, Mathias Lepoutre, and Bernd Bickel. 2016. Computational Design of Stable Planar-rod Structures. *ACM Trans. Graph.* 35, 4, Article 86 (July 2016), 11 pages. <https://doi.org/10.1145/2897824.2925978>
- Joseph S. B. Mitchell, David M. Mount, and Christos H. Papadimitriou. 1987. The Discrete Geodesic Problem. *SIAM J. Comput.* 16, 4 (Aug. 1987), 647–668. <https://doi.org/10.1137/0216045>
- Vidya Narayanan, Lea Albaugh, Jessica Hodgins, Stelian Coros, and James Mccann. 2018. Automatic Machine Knitting of 3D Meshes. *ACM Trans. Graph.* 37, 3, Article 35 (Aug. 2018), 15 pages. <https://doi.org/10.1145/3186265>
- M. Nieser, J. Palacios, K. Polthier, and E. Zhang. 2012. Hexagonal Global Parameterization of Arbitrary Surfaces. *IEEE Transactions on Visualization and Computer Graphics* 18, 6 (June 2012), 865–878. <https://doi.org/10.1109/TVCG.2011.118>
- Jonathan Palacios and Eugene Zhang. 2007. Rotational Symmetry Field Design on Surfaces. *ACM Trans. Graph.* 26, 3, Article 55 (July 2007), 10 pages. <https://doi.org/10.1145/1276377.1276446>
- Daniele Panozzo, Enrico Puppo, Marco Tarini, and Olga Sorkine-Hornung. 2014. Frame Fields: Anisotropic and Non-orthogonal Cross Fields. *ACM Trans. Graph.* 33, 4, Article 134 (July 2014), 11 pages. <https://doi.org/10.1145/2601097.2601179>
- Jesús Pérez, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, José A. Canabal, Robert Sumner, and Miguel A. Otaduy. 2015. Design and Fabrication of Flexible Rod Meshes. *ACM Trans. Graph.* 34, 4, Article 138 (July 2015), 12 pages. <https://doi.org/10.1145/2766998>
- Henri Poincaré. 1905. Sur Les Lignes Geodesiques Des Surfaces Convexes. *Trans. Amer. Math. Soc.* 6, 3 (1905), 237–274. <http://www.jstor.org/stable/1986219>
- S. Poljak and Z. Tuzs. 1994. *The Max-cut Problem: A Survey*. Institute of Mathematics, Academia Sinica, Taipei, Taiwan, China. <https://books.google.com/books?id=AaZMwEACAAJ>
- Konrad Polthier and Eike Preuß. 2003. Identifying Vector Field Singularities Using a Discrete Hodge Decomposition. In *Visualization and Mathematics III*, Hans-Christian Hege and Konrad Polthier (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 113–134.
- Konrad Polthier and Markus Schmies. 2006. Straightest Geodesics on Polyhedral Surfaces. In *ACM SIGGRAPH 2006 Courses (SIGGRAPH '06)*. ACM, New York, NY, USA, 30–38. <https://doi.org/10.1145/1185657.1185664>
- Mariana Popescu, Matthias Rippmann, Tom Van Mele, and Philippe Block. 2018. *Automated Generation of Knit Patterns for Non-developable Surfaces*. Springer Singapore, Singapore, 271–284. [https://doi.org/10.1007/978-981-10-6611-5\\_24](https://doi.org/10.1007/978-981-10-6611-5_24)
- Helmut Pottmann, Qixing Huang, Bailin Deng, Alexander Schiftner, Martin Kilian, Leonidas Guibas, and Johannes Wallner. 2010. Geodesic Patterns. *ACM Trans. Graph.* 29, 4, Article 43 (July 2010), 10 pages. <http://doi.acm.org/10.1145/1778765.1778780>
- Martin Puryear. 1998. Brunhilde. Cedar and rattan.
- Qmechanic. 2018. Is every vector field locally integrable up to a rescaling? Mathematics Stack Exchange. URL:<https://math.stackexchange.com/q/2716593> (version: 2018-04-10).
- Nicolas Ray, Wan Chiu Li, Bruno Lévy, Alla Sheffer, and Pierre Alliez. 2006. Periodic Global Parameterization. *ACM Trans. Graph.* 25, 4 (Oct. 2006), 1460–1485. <https://doi.org/10.1145/1183287.1183297>
- Nicolas Ray and Dmitry Sokolov. 2014. Robust Polylines Tracing for N-Symmetry Direction Field on Triangulated Surfaces. *ACM Trans. Graph.* 33, 3, Article 30 (June 2014), 11 pages. <https://doi.org/10.1145/2602145>
- Nicolas Ray, Bruno Vallet, Laurent Alonso, and Bruno Levy. 2009. Geometry-aware Direction Field Processing. *ACM Trans. Graph.* 29, 1, Article 1 (Dec. 2009), 11 pages. <https://doi.org/10.1145/1640443.1640444>
- Tristan Rivière and Sylvia Serfaty. 2001. Limiting domain wall energy for a problem related to micromagnetics. *Communications on Pure and Applied Mathematics* 54, 3 (2001), 294–338. [https://doi.org/10.1002/1097-0312\(200103\)54:3<294::AID-CPA2>3.0.CO;2-S](https://doi.org/10.1002/1097-0312(200103)54:3<294::AID-CPA2>3.0.CO;2-S) arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/1097-0312%28200103%2954%3A3%3C294%3A%3AAID-CPA2%3E3.0.CO%3B2-S>
- Lawrence Roy, Prashant Kumar, Sanaz Golbabaee, Yue Zhang, and Eugene Zhang. 2018. Interactive Design and Visualization of Branched Covering Spaces. *IEEE Trans. Vis. Comput. Graph.* 24, 1 (2018), 843–852. <https://doi.org/10.1109/TVCG.2017.2744038>
- Zhongwei Shen, Jin Huang, Wei Chen, and Hujun Bao. 2015. Geometrically Exact Simulation of Inextensible Ribbon. *Comput. Graph. Forum* 34, 7 (Oct. 2015), 145–154. <https://doi.org/10.1111/cgf.12753>
- Mélina Skouras, Stelian Coros, Eitan Grinspun, and Bernhard Thomaszewski. 2015. Interactive Surface Design with Interlocking Elements. *ACM Trans. Graph.* 34, 6, Article 224 (Oct. 2015), 224:1–224:7 pages. <https://doi.org/10.1145/2816795.2818128>
- Peng Song, Chi-Wing Fu, Prashant Goswami, Jianmin Zheng, Niloy J. Mitra, and Daniel Cohen-Or. 2013. Reciprocal Frame Structures Made Easy. *ACM Trans. Graph.* 32, 4, Article 94 (July 2013), 13 pages. <https://doi.org/10.1145/2461912.2461915>
- Vitaly Surazhsky, Tatiana Surazhsky, Danil Kirsanov, Steven J. Gortler, and Hugues Hoppe. 2005. Fast Exact and Approximate Geodesics on Meshes. *ACM Trans. Graph.* 24, 3 (July 2005), 553–560. <https://doi.org/10.1145/1073204.1073228>
- Masahito Takezawa, Takuma Imai, Kentaro Shida, and Takashi Maekawa. 2016. Fabrication of Freeform Objects by Principal Strips. *ACM Trans. Graph.* 35, 6, Article 225 (Nov. 2016), 12 pages. <https://doi.org/10.1145/2980179.2982406>
- Ye Tao, Nannan Lu, Caowei Zhang, Guanyun Wang, Cheng Yao, and Fangtian Ying. 2016. CompuWoven: A Computer-Aided Fabrication Approach to Hand-Woven Craft. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. ACM, New York, NY, USA, 2328–2333. <https://doi.org/10.1145/2851581.2892293>
- Ye Tao, Guanyun Wang, Caowei Zhang, Nannan Lu, Xiaoliang Zhang, Cheng Yao, and Fangtian Ying. 2017. WeaveMesh: A Low-Fidelity and Low-Cost Prototyping Approach for 3D Models Created by Flexible Assembly. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 509–518. <https://doi.org/10.1145/3025453.3025699>
- Hugues Vandeparre, Miguel Piñeirua, Fabian Brau, Benoit Roman, José Bico, Cyprien Gay, Wenzhong Bao, Chun Ning Lau, Pedro M. Reis, and Pascal Damman. 2011. Wrinkling Hierarchy in Constrained Thin Sheets from Suspended Graphene to Curtains. *Phys. Rev. Lett.* 106 (Jun 2011), 224301. Issue 22. <https://doi.org/10.1103/PhysRevLett.106.224301>
- Amir Vaxman, Marcel Campen, Olga Diamanti, Daniele Panozzo, David Bommes, Klaus Hildebrandt, and Mirela Ben-Chen. 2016. Directional Field Synthesis, Design, and Processing. In *Proceedings of the 37th Annual Conference of the European Association for Computer Graphics: State of the Art Reports (EG '16)*. Eurographics Association, Goslar Germany, Germany, 545–572. <https://doi.org/10.1111/cgf.12864>
- R. Viertel and B. Osting. 2019. An Approach to Quad Meshing Based on Harmonic Cross-Valued Maps and the Ginzburg-Landau Theory. *SIAM Journal on Scientific Computing* 41, 1 (2019), A452–A479. <https://doi.org/10.1137/17M1142703> arXiv:<https://doi.org/10.1137/17M1142703>
- Kui Wu, Xifeng Gao, Zachary Ferguson, Daniele Panozzo, and Cem Yuksel. 2018. Stitch Meshing. *ACM Trans. Graph.* 37, 4 (2018).
- Qing Xing, Gabriel Esquivel, Ergun Akleman, Jianer Chen, and Jonathan Gross. 2011. Band Decomposition of 2-manifold Meshes for Physical Construction of Large Structures. In *ACM SIGGRAPH 2011 Posters (SIGGRAPH '11)*. ACM, New York, NY, USA, Article 58, 1 pages. <https://doi.org/10.1145/2037715.2037781>
- Alexander Yampolsky. 2005. On extrinsic geometry of unit normal vector fields of Riemannian hyperfoliations. (2005). arXiv:arXiv:math/0503566
- Cem Yuksel, Jonathan M. Kaldor, Doug L. James, and Steve Marschner. 2012. Stitch Meshes for Modeling Knitted Clothing with Yarn-level Detail. *ACM Trans. Graph.* 31, 4, Article 37 (July 2012), 12 pages. <https://doi.org/10.1145/2185520.2185533>
- Jonas Zehnder, Stelian Coros, and Bernhard Thomaszewski. 2016. Designing Structurally-sound Ornamental Curve Networks. *ACM Trans. Graph.* 35, 4, Article 99 (July 2016), 10 pages. <https://doi.org/10.1145/2897824.2925888>
- Eugene Zhang, Konstantin Mischaikow, and Greg Turk. 2006. Vector Field Design on Surfaces. *ACM Trans. Graph.* 25, 4 (Oct. 2006), 1294–1326. <https://doi.org/10.1145/1183287.1183290>
- Mateusz Zwierzycki, Petras Vestartas, Mary Katherine Heinrich, and Phil Ayres. 2017. High Resolution Representation and Simulation of Braiding Patterns. In *Acadia 2017: Disciplines and Disruption*, Vol. MA 2-4. Acadia Publishing Company, Boston, MA, 670–679.

## A SMOOTH SOLUTIONS TO GEODESIC VARIATIONAL PROBLEM

Here we prove that in the smooth setting, the solutions  $(\hat{w}, \delta)$  to equation 4 are precisely geodesic fields, with  $\delta = 0$ . We begin with the observation that at optimality, all of the following must be true:

- The two fields  $\hat{w}$  and  $\delta$  are parallel, so that  $\delta = s\hat{w}$  for some function  $s : M \rightarrow \mathbb{R}$ ;
- $\int_M s \, dA = 0$ , since otherwise  $\delta$  can be globally rescaled to decrease the objective function, while maintaining all constraints;
- for any curl-free vector field  $v$ ,  $\int_M \langle v, \delta \rangle \, dA = 0$ , since otherwise the  $v$  component of  $\delta$  can be subtracted from  $\delta$  to decrease the objective function.

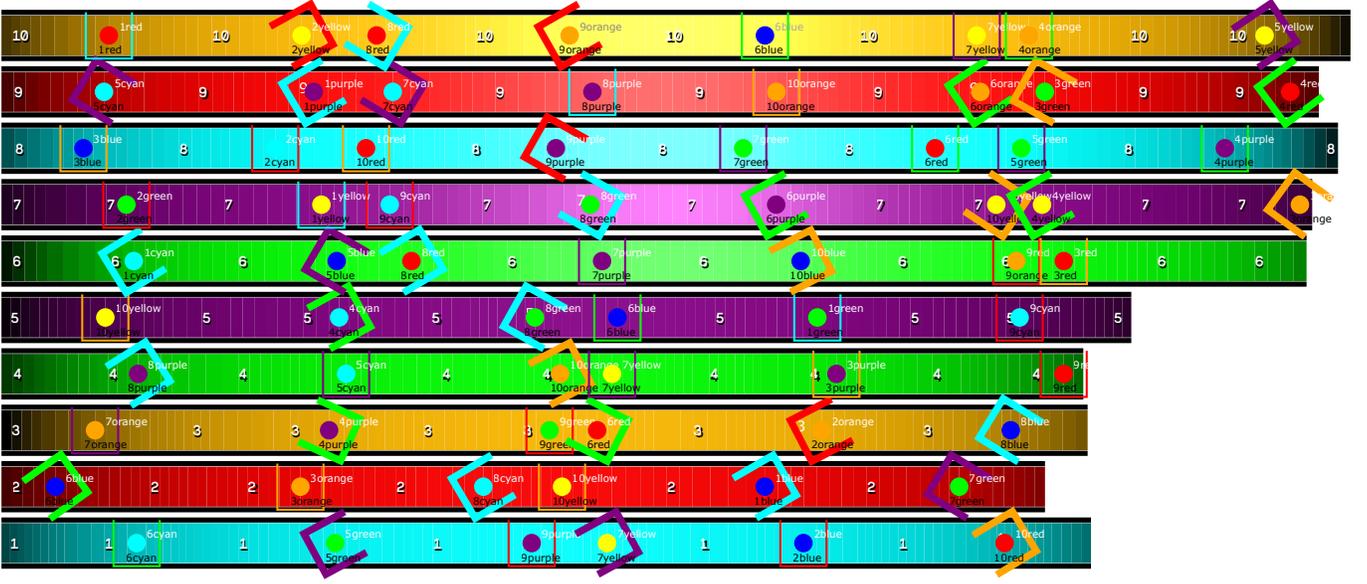


Fig. 25. Try assembling this design file for a cone with angle 60 degrees

Put together, these facts imply that at optimality,  $\delta = s\hat{w}$  is orthogonal to  $\hat{w} + \delta = (1+s)\hat{w}$  since the latter is curl-free:

$$0 = \int_M \langle (1+s)\hat{w}, s\hat{w} \rangle dA = \int_M s^2 dA$$

which implies  $s = 0$ . Thus  $\delta = 0$  and  $\hat{w}$  is geodesic.

## B GEODESIC FIELD OPTIMIZATION

In code, we represent vector fields on faces of  $\mathcal{T}$  in reduced coordinates  $\mathbf{u} \in \mathbb{R}^{2|\mathcal{F}|}$ , where  $\mathbf{u}_i$  are the two barycentric coordinates of  $\mathbf{w}_i$  in the coordinate system of face  $i$ . We will denote by  $\bar{\mathbf{u}}_i \in \mathbb{R}^3$  the inclusion of  $\mathbf{u}_i$  into ambient space.

Throughout the appendices we will make use of the following discrete operators:

- the linear discrete curl operator  $C_{|\mathcal{E}_{\text{int}}| \times 2|\mathcal{F}|}$  acting on vector fields in the barycentric basis, defined as in equation 3. Here  $\mathcal{E}_{\text{int}}$  is the set of interior (non-boundary) edges;
- the  $L_2$  inner product on functions on faces  $M_{\mathcal{F}}$ , an  $\mathcal{F} \times \mathcal{F}$  diagonal matrix with entries equal to the areas  $\mathcal{A}_i$  of triangles  $i$ ;
- the corresponding  $L_2$  inner product on vector fields  $M_{\mathcal{F}}^v$  defined by

$$\mathbf{u}^T M_{\mathcal{F}}^v \mathbf{u} = \sum_{i \in \mathcal{F}} \mathcal{A}_i \|\bar{\mathbf{u}}_i\|^2;$$

- the Laplacian on faces  $L_{\mathcal{F}}$ , which we discretize using the reciprocal cotan weights of the ordinary vertex Laplacian, e.g.

$$(L_{\mathcal{F}}\mathbf{s})_i = \sum_{j \sim i} \frac{s_j - s_i}{\omega_{ij}},$$

where the sum is over all face neighbors of face  $i$  and  $\omega_{ij}$  is the cotan weight of the shared edge;

- the Dirichlet energy on vector fields  $\mathbf{u}$ , given by

$$\|\nabla \mathbf{u}\|^2 = \sum_{e_{ij} \in \mathcal{E}_{\text{int}}} \frac{1}{\omega_{ij}} \left\| \overline{T_{j \leftarrow i} \mathbf{u}_i} - \bar{\mathbf{u}}_j \right\|^2,$$

where  $T_{j \leftarrow i}$  is the transport operator from face  $i$  to face  $j$ , as in equation (2). We write the Dirichlet energy in matrix form as

$$\|\nabla \mathbf{u}\|^2 = -\mathbf{u}^T L_{\mathcal{F}}^v \mathbf{u}.$$

In terms of these operators, the variational problem (5) becomes

$$\begin{aligned} & C(\mathbf{u} + \delta) = 0, \\ \min_{\delta, \mathbf{u}} & \frac{1}{2} \delta^T M_{\mathcal{F}}^v \delta - \frac{\lambda}{2} [\mathbf{u} + \delta]^T L_{\mathcal{F}}^v [\mathbf{u} + \delta] \quad \text{s.t.} \quad \|\bar{\mathbf{u}}_i\| = 1, \quad (9) \\ & \mathbf{u}_i = \mathbf{u}_i^{\mathcal{H}} \quad \forall i \in \mathcal{H} \end{aligned}$$

As described in Algorithm 1, we solve this problem in alternating fashion: (1) fix  $\mathbf{u}$  and solve for  $\delta$ , which amounts to a convex quadratic program; (2) fix  $\delta$  and update  $\mathbf{u}$ . Both steps decrease the energy (9) while maintaining the constraints, and hence the alternation will converge.

**$\delta$  Update Step.** Suppose we have current values  $\delta^i, \mathbf{u}^i$  for the optimization variables that satisfy all constraints. Fixing  $\mathbf{u}^i$  and finding  $\delta^{i+1/2}$  which minimizes equation (9) amounts to linearly-constrained least squares, since the unit-norm constraint on  $\mathbf{u}_i$  is irrelevant. We introduce  $|\mathcal{E}_{\text{int}}|$  Lagrange multipliers  $\boldsymbol{\mu}$  (the handle constraints can be incorporated either by deleting the constrained  $\delta_i$  from the optimization, or by introducing additional, trivial constraints and Lagrange multipliers) and solve

$$\tilde{\delta} = \arg \min_{\delta, \boldsymbol{\mu}} \frac{1}{2} \delta^T M_{\mathcal{F}}^v \delta + \frac{\lambda}{2} [\mathbf{u}^i + \delta]^T L_{\mathcal{F}}^v [\mathbf{u}^i + \delta] + \boldsymbol{\mu}^T C [\mathbf{u}^i + \delta].$$

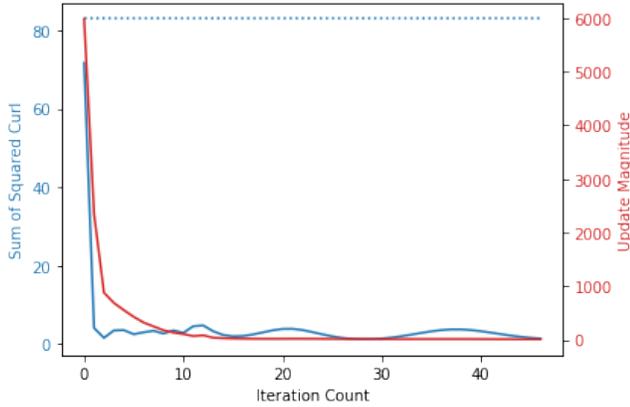


Fig. 26. Convergence of our algorithm on the sphere example in Figure 7 (with  $\lambda = 100$ ). Both the total curl (blue) and change in  $\hat{\mathbf{w}}$  per iteration (red) decreases quickly with increasing number of iterations. The dotted blue line is the total curl of a globally optimal vector field [Knöppel et al. 2013], which we use as input to Algorithm 1. As illustrated in Figure 9, the curl can be further reduced on many examples by allowing  $\lambda \rightarrow 0$  after converging on an initial smooth solution.

The KKT conditions of the above problem form a linear system in terms of  $\delta$  and  $\mu$ :

$$\left[ \begin{array}{c|c} M_{\mathcal{F}}^{\mathbf{v}} + \lambda L_{\mathcal{F}}^{\mathbf{v}} & C^T \\ \hline C & 0 \end{array} \right] \begin{bmatrix} \delta \\ \mu \end{bmatrix} = \begin{bmatrix} -\lambda L_{\mathcal{F}}^{\mathbf{v}}(\mathbf{u}^i + \delta) \\ -C(\mathbf{u}^i) \end{bmatrix}$$

which we solve using sparse QR decomposition.

**u Update Step.** In the second alternation step, we set

$$\mathbf{u}_j^{i+1} = \frac{\mathbf{u}_j^i + \tilde{\delta}_j}{\|\mathbf{u}_j^i + \tilde{\delta}_j\|} \quad \delta_j^{i+1} = -\mathbf{u}_j^{i+1} + \mathbf{u}_j^i + \tilde{\delta}_j.$$

Notice that  $\mathbf{u}^i + \tilde{\delta} = \mathbf{u}^{i+1} + \delta^{i+1}$ , while the norm of  $\delta^{i+1}$  has decreased, so that this step decreases (9) while maintaining both constraints.

### B.1 Convergence

Figure 26 shows convergence of our algorithm on the sphere example considered in Figure 7 (with  $\lambda = 100$ ). We plot the change in  $\mathbf{w}$  between iterations, as well as the total curl  $\int_M \|\nabla \times \hat{\mathbf{w}}\|$  over the surface. The curl rapidly decreases after a few iterations of the above alternation (it never reaches exactly zero due to both discretization error in estimating the total curl, and the influence of the regularization term  $\|\nabla(\hat{\mathbf{w}} + \delta)\|^2$ ).

We measure curl of a vector field  $\mathbf{w}$  on each face of a mesh as follows: with  $e_{ij}$  the edge vector shared by faces  $i$  and  $j$ , oriented consistently with face  $i$ , we compute

$$(\nabla \times \mathbf{w})_i = \frac{1}{\mathcal{A}_i} \sum_{j \sim i} (\mathbf{w}_j - \mathbf{w}_i) \cdot e_{ij},$$

where the sum is over faces  $j$  neighboring face  $i$ .

## C COMPUTING AN INITIAL RESCALING

Here we provide details regarding the numerical recovery of  $\mathbf{s}$  that minimizes Equation (7) by solving a generalized eigenvalue problem.

**ALGORITHM 2:** Inverse power iteration for finding  $\mathbf{s}$ .

```

 $\mathbf{x}_0 \leftarrow \text{rand}();$ 
 $\mathbf{x}_0 \leftarrow \mathbf{x}_0 / \|\mathbf{x}_0\|_{A_2};$ 
for  $i = 0, \dots, i_{\max}$  do
   $\tilde{\mathbf{x}}_{i+1} \leftarrow (A_1 + A_2)^{-1} A_2 \mathbf{x}_i;$ 
   $\mu \leftarrow (BB^T)^{-1} B \tilde{\mathbf{x}}_{i+1};$  // project onto nullspace of  $B$ 
   $\mathbf{x}_{i+1} \leftarrow \tilde{\mathbf{x}}_{i+1} - B^T \mu;$ 
   $\mathbf{x}_{i+1} \leftarrow \mathbf{x}_{i+1} / \|\mathbf{x}_{i+1}\|_{A_2};$ 
end

```

Equation (7) is equivalent to the constrained Rayleigh quotient problem

$$\min_{\mathbf{x}} \frac{\mathbf{x}^T A_1 \mathbf{x}}{\mathbf{x}^T A_2 \mathbf{x}} \quad \text{s.t.} \quad B\mathbf{x} = 0$$

with  $\mathbf{x} = \begin{bmatrix} \delta & \mathbf{s} \end{bmatrix}^T$  and

$$A_1 = \begin{bmatrix} M_{\mathcal{F}}^{\mathbf{v}} & \\ & -\mu L_{\mathcal{F}} \end{bmatrix}, \quad A_2 = \begin{bmatrix} M_{\mathcal{F}}^{\mathbf{v}} & \\ & M_{\mathcal{F}} \end{bmatrix},$$

in the notation of Appendix B.

Concretely, note that  $A_1$  is positive-semidefinite and  $A_2$  is positive-definite. As usual, the solution to the Rayleigh quotient optimization problem is the generalized eigenvector of  $A_1|_B$  with respect to  $A_2|_B$ , where  $A_1|_B$  is the restriction of  $A_1$  to the nullspace of  $B$ . We solve this optimization problem using inverse power iteration, as summarized in Algorithm 2. We add  $A_2$  to  $A_1$  in the algorithm, as this addition shifts the spectrum of  $A_1$  without affecting its eigenvectors, while ensuring that  $A_1 + A_2$  is invertible.

If the surface  $\mathcal{T}$  consists of multiple connected components, algorithm 2 must be run separately on each component, as otherwise the solution will set  $\mathbf{s} = 0$  on all but one of the components. Finally, both matrices  $A_1 + A_2$  and  $BB^T$  can and should be Cholesky pre-factorized before the power iteration loop.

Table 2. We emphasize two differences between Equations 5 and 7.

|      | Constraint Space<br>(+ integrability)                            | Solution Space   |
|------|--|--|
| Eq 5 | Unit norm $\ \cdot\ ^2$<br>per face (non-convex)                 | Smooth $S^1$ valued<br>parameterization of<br>unit field $\hat{\mathbf{w}}$ per face |
| Eq 7 | $\ \mathbf{s}\mathbf{w}^\perp\ ^2 = C$<br>integrated (quadratic) | Smooth $\mathbb{R}^+$ field $\mathbf{s}$<br>per face                                 |

## D 6-ROSY OPTIMIZATION

We use a globally optimal [Knöppel et al. 2013] 6-RoSy for the purposes of initializing branched cover singularities (Section 5.1). We briefly describe how to compute such a RoSy field, in the interest of making this paper more self-contained.

*Representing 6-RoSys.* Let  $\mathbf{v}$  denote an arbitrary *reference* vector field on the faces  $\mathcal{F}$  of  $\mathcal{T}$ . Then the six vectors  $\{\hat{\mathbf{w}}_i^j\}_{j=1}^6$  of a 6-RoSy on face  $i$  are related by a rotation of  $\pi/3$  radians by the surface normal. Given a vector  $\mathbf{w}$  on face  $i$ , define  $R_i(\mathbf{w})$  to be the rotation in the tangent plane of face  $i$  mapping  $\mathbf{v}_i$  to  $\mathbf{w}$ ; the RoSy can then be compactly represented by a *representative vector field*  $\hat{\mathbf{w}}^\star$  with

$$\hat{\mathbf{w}}_i^\star = R_i(\mathbf{w}_i^j)^6 \mathbf{v}_i;$$

here the rotation on the right is raised to the sixth power. Note that the representative vector field for a RoSy is unique, given  $\mathbf{v}$ , and does not depend on the labeling of the six vectors in the RoSy on a face.

*Dirichlet Energy.* For two adjacent faces  $i, j$ , denote by  $T_{j \leftarrow i}^V$  the parallel transport operator on RoSys, defined by

$$R_j(T_{j \leftarrow i} \mathbf{w}) \mathbf{v}_j = T_{j \leftarrow i} R_i(\mathbf{w}) \mathbf{v}_i$$

for all tangent vectors  $\mathbf{w}$  on face  $i$ . The map  $T_{j \leftarrow i}^V$  is linear and depends only on  $\mathcal{T}$  and  $\mathbf{v}$ , and can be precomputed as a matrix on each edge of  $\mathcal{T}$ . The Dirichlet energy on 6-RoSy representative vector fields is then

$$\|\nabla \hat{\mathbf{w}}^\star\|^2 = \sum_{e_{ij} \in \mathcal{E}_{\text{int}}} \frac{1}{\omega_{ij}} \left\| \overline{T_{j \leftarrow i}^V \hat{\mathbf{w}}_i^\star} - \overline{\hat{\mathbf{w}}_j^\star} \right\|^2,$$

in direct analogy to the Dirichlet energy in section B.

*Globally Optimal RoSys.* Following Knoppel et al. [2013], finding an as-smooth-as-possible 6-RoSy on a surface can be made well-posed by relaxing the unit magnitude condition on the representative vector field:

$$\arg \min_{\mathbf{w}^\star} \|\nabla \mathbf{w}^\star\|^2 \quad \text{s.t.} \quad \|\mathbf{w}^\star\|^2 = 1,$$

whose solution is given by a generalized eigenvector problem.

## E WEAVE SIMULATION AND FINE-TUNING

Although geodesic foliations are good heuristics for weave patterns that will stably approximate a shape when woven, we perform a physical simulation to fine-tune the pattern design and to predict the equilibrium shape of the weave (and in particular, detect if the weave will remain stable under its own weight) after fabrication. In this section, we describe this stage of our pipeline, where we solve a global statics problem to relax the weave to equilibrium while staying as close as possible to the design goal. At a high level, we improve the weave in two steps:

- we allow ribbons to *slide on the surface* of the target mesh, while remaining constrained to lie tangent to the target surface. This sliding corrects small artifacts in the field design or integral curve extraction steps, and reduces strain in the weave, particularly near singularities where many ribbons terminate.
- we relax the weave to static equilibrium. This step gives the user a preview of what shape the weave will assume when fabricated. Both steps make use of an elastic rod simulation of the weave. We assume that each ribbon of the weave has some user-specified thickness  $h$  and width  $w$ , with  $w > h$ , and represent the weave as an anisotropic thin elastic rod with rectangular cross-section. We base

our simulation on the formulation of Bergou et al. [2010; 2008], which models elastic rods using a Kirchhoff kinematic model, with:

- a piecewise-linear *centerline* specified by centerline vertices  $V_c = \{\mathbf{p}_1, \dots, \mathbf{p}_{|V_c|}\}$ ;
- a unit *reference director*  $\mathbf{d}_{i+1/2}$  on every line segment connecting vertices  $\mathbf{p}_i$  and  $\mathbf{p}_{i+1}$ , and orthogonal to that segment;
- an angle  $\theta_{i+1/2}$  per line segment, specifying the orientation of the segment, with the thickness direction  $\tau_{i+1/2}$  of the ribbon (normal to the width direction) given by

$$\left( \cos \theta_{i+1/2} \right) \mathbf{d}_{i+1/2} + \left( \sin \theta_{i+1/2} \right) \hat{\mathbf{t}}_{i+1/2} \times \mathbf{d}_{i+1/2},$$

where  $\hat{\mathbf{t}}_{i+1/2} = (\mathbf{p}_{i+1} - \mathbf{p}_i) / \|\mathbf{p}_{i+1} - \mathbf{p}_i\|$  is the unit vector tangent to the ribbon.

We chose Bergou et al's method as it was thoroughly validated by its authors, and has been successfully applied to a variety of problems in both physics [Brun et al. 2015] and computational fabrication [Pérez et al. 2015].

We convert the isolines extracted in section 5 into elastic rods by initially setting  $\theta_{i+1/2} = 0$  and  $\mathbf{d}_{i+1/2}$  equal to the surface normal of  $\mathcal{T}$  at the midpoint of each rod segment. We also store a rest length  $\ell_{i+1/2}$  on each segment, equal to  $\|\mathbf{p}_{i+1} - \mathbf{p}_i\|$  at the start of this step of the pipeline.

### E.1 Fine-tuning By Sliding

To improve the weave, we allow ribbons to slide over each other while remaining close to the target surface. Let  $E_{\text{int}}$  be the total internal potential energy of the weave; to this elastic energy we add an energy that penalizes movement of a ribbon away from the surface, similar to the classic technique of implementing frictional contact using anchored springs [Howard and Kumar 1993], or point-to-plane energies to solve ICP registration problems. Denote by  $\mathbf{a}_i$  an *anchor point* on  $\mathcal{T}$  associated to rod vertex  $i$ , and by  $\hat{\mathbf{n}}_i$  the surface normal at  $\mathbf{a}_i$ . Our sliding energy for a rod is then

$$E_{\text{slide}} = \sum_{i=1}^{|V_c|} \left[ \frac{\kappa_n}{2} \|\hat{\mathbf{n}}_i \cdot (\mathbf{p}_i - \mathbf{a}_i)\|^2 + \frac{\kappa_t}{2} \left\| \left( I - \hat{\mathbf{n}}_i \hat{\mathbf{n}}_i^T \right) (\mathbf{p}_i - \mathbf{a}_i) \right\|^2 \right],$$

where the first term penalizes normal motion of the rod away from  $\mathcal{T}$ , and the second, tangential sliding (we use  $\kappa_n = 1000$  and  $\kappa_t = 1$ ). We alternate (i) solving

$$\arg \min_{\mathbf{p}, \theta} \sum_{\text{rods}} (E_{\text{int}} + E_{\text{slide}})$$

using Gauss-Newton optimization (here the  $\mathbf{a}_i$  are kept fixed), and (ii) recomputing each  $\mathbf{a}_i$  by projection  $\mathbf{p}_i$  onto  $\mathcal{T}$ . The optimization step allows ribbons to slide along the surface in order to relieve internal stress; the  $\kappa_n$  term serves as a soft constraint restricting normal motion, and  $\kappa_t$  term regularizes the optimization by preventing rods from sliding too much within each iteration.

### E.2 Relaxing The Weave

We perform a final simulation (without any artificial penalty forces keeping the ribbons near  $\mathcal{T}$ ) to predict the equilibrium shape of the weave pattern. If fabricated, the weave will hold together in this predicted shape. For closed surfaces, or open surfaces with rigid boundary, the equilibrium shape is almost always very close to the

input target shape, though we did encounter a few failure cases where the weave relaxes to a significantly different shape when the boundary is open (see Figure 22).

We assume that contact between ribbons is unilateral: that either the over-under weaving of the ribbons is sufficient to hold the weave together via stiction, or that ribbons are otherwise fastened together at their crossings. Contact forces at the crossings balance residual internal forces within the ribbons, yielding a stable curved shape: without contact forces each ribbon would unravel into a straight, flat strip. We simulate contact constraints by augmenting the internal energies of Bergou et al. [2008] with a penalty potential per ribbon crossing. In modelling this potential, we account for (i) ribbons weaving over and under each other, so that when they are in contact their centerlines are distance  $h$  apart at the crossing; and (ii) pairs of ribbons must lie flush when they cross, and so must have parallel normals at the contact point.<sup>5</sup>

Suppose a crossing is present on segments  $i + 1/2$  and  $j + 1/2$  on rods  $a$  and  $b$ , so that the centerline positions closest to the point of contact are given by

$$\mathbf{c}^a = (1 - \alpha^a)\mathbf{p}_i^a + \alpha^a\mathbf{p}_{i+1}^a; \quad \mathbf{c}^b = (1 - \alpha^b)\mathbf{p}_j^b + \alpha^b\mathbf{p}_{j+1}^b$$

for barycentric coordinates  $\alpha^a$  and  $\alpha^b$ , where the superscripts  $a, b$  are used to denote quantities associated to the corresponding rod. Then for this crossing we use contact penalty

$$E_c = \gamma_{\text{pos}} \left( \left\| \mathbf{c}^a + \epsilon h \boldsymbol{\tau}_{i+1/2}^a - \mathbf{c}^b \right\|^2 + \left\| \mathbf{c}^b - \epsilon h \boldsymbol{\tau}_{j+1/2}^b - \mathbf{c}^a \right\|^2 \right) \\ + \gamma_{\text{orient}} \left\| \angle \left( \boldsymbol{\tau}_{i+1/2}^a, \boldsymbol{\tau}_{j+1/2}^b \right) \right\|^2$$

for stiffness parameters  $\gamma_{\text{pos}}$  and  $\gamma_{\text{orient}}$ , and for  $\epsilon \in \{1, -1\}$  encoding the over-under assignment. We use  $\gamma_{\text{pos}} = 1000$  and  $\gamma_{\text{orient}} = 10^6$  for all of our examples.

*Numerical Optimization.* Static equilibria of the weave are characterized by extrema of the weave's total potential energy, including penalty terms,

$$E = \sum_{\text{rods } r} E_{\text{int}} + \sum_{\text{contacts } c} E_c. \quad (10)$$

There are always multiple such extrema, since  $E$  is invariant under rigid motions of the weave. We therefore regularize Equation (10) by constraining the weave center of mass and rigid orientation to remain constant during optimization.

Although every term in  $E$  is quite nonlinear, we use Gauss-Newton optimization with line search to solve equation (10), and in our experience it quickly converges to a weave equilibrium state. There is one minor complication: during the line search, the directors  $\mathbf{d}_{i+1/2}$  stop being normal to their corresponding rod segments, so that the estimate of the post-step energy in the line search is incorrectly optimistic. A correct line search must parallel-transport (see Bergou et al. [2008] for details) the directors onto the predicted new centerline segments when evaluating the suitability of a step.

<sup>5</sup>We note that this model does nothing to ensure that the target weave pattern is actually fabricable (as the width of the ribbons imposes a constraint on the weaving pattern that we do not account for).

## F EXAMPLE WEAVE PATTERN

We have included in Figure 25 a simple weave pattern generated using our pipeline; we encourage the reader to assemble the object (a cone with  $60^\circ$  cone angle) themselves. Cut out the ribbons, and fasten ribbons to each other at the indicated crossing points, with indicated orientation. All ribbons should be oriented so that the colored side of the paper points out: the ribbon with thicker guide marker should be woven under the ribbon with thinner marker.