Chebyshev Nets from Commuting PolyVector Fields

ANDREW O. SAGEMAN-FURNAS, Technical University of Berlin
ALBERT CHERN, Technical University of Berlin
MIRELA BEN-CHEN, Technion-Israel Institute of Technology
AMIR VAXMAN, Utrecht University

Fig. 1. Given a target surface, our fully automatic method computes a commuting unit length PolyVector guiding field with optimally placed singularities (left) that are adapted for integrating into an approximate Chebyshev net where all edge lengths are equal (middle). This global Chebyshev net with singularities is physically realized with an interwoven net of nearly inextensible yarns (right).

We propose a method for computing global Chebyshev nets on triangular meshes. We formulate the corresponding global parameterization problem in terms of *commuting* PolyVector fields, and design an efficient optimization method to solve it. We compute, for the first time, Chebyshev nets with automatically-placed singularities, and demonstrate the realizability of our approach using real material.

CCS Concepts: • **Computing methodologies → Mesh geometry models**.

Additional Key Words and Phrases: Chebyshev nets, Directional fields, PolyVector fields, Ginzburg–Landau, Lie brackets

ACM Reference Format:
Andrew O. Sageman-Furnas, Albert Chern, Mirela Ben-Chen, and Amir Vaxman . 2019. Chebyshev Nets from Commuting PolyVector Fields. *ACM Trans. Graph.* 38, 6, Article 172 (November 2019), 16 pages. https://doi.org/10.1145/3355089.3356564

Authors' addresses: Andrew O. Sageman-Furnas Technical University of Berlin, aosafu@math.tu-berlin.de; Albert Chern Technical University of Berlin, chern@math.tu-berlin.de; Mirela Ben-Chen Technion-Israel Institute of Technology, mirela@cs.technion.ac.il; Amir Vaxman Utrecht University, a.vaxman@uu.nl.

0730-0301/2019/11-ART172
https://doi.org/10.1145/3355089.3356564

1 INTRODUCTION

Chebyshev nets model two-dimensional materials that are built from grids of flexible, nearly inextensible rods. Everyday examples range from large architectural gridshells to fruit packaging (Fig. 2). Important applications include static reinforcement, as in composite materials for industrial design, and dynamic reinforcement, as in medical meshes such as stents. Finding a Chebyshev net that approximates an arbitrary surface is essential to these applications. However, this task is well known to be challenging due to the anisotropic inextensibility of such nets.

The inextensibility along the two rod directions of a Chebyshev net induces geometric obstructions to encoding large regions of curvature. These geometric constraints lead to places where the angle between adjacent rods collapses, resulting in degenerate configurations. Such configurations must be avoided because physical rods have finite thickness, resulting in a minimum and maximum angle that can be achieved in fabrication. One can locally propagate a Chebyshev net from initial curves along a target surface, but generically this leads to degeneracies, which prevent further propagation. Previous works have either presented algorithms to compute Chebyshev nets on local patches of surfaces, or overcame degeneracies by manually choosing initial data and then having user-aided guidance or heuristically inserting singularities.

Our goal is to design an algorithm that *automatically* computes a *global* Chebyshev net, where the singularities arise automatically. *Discrete Chebyshev nets* are quad meshes where all edge lengths are

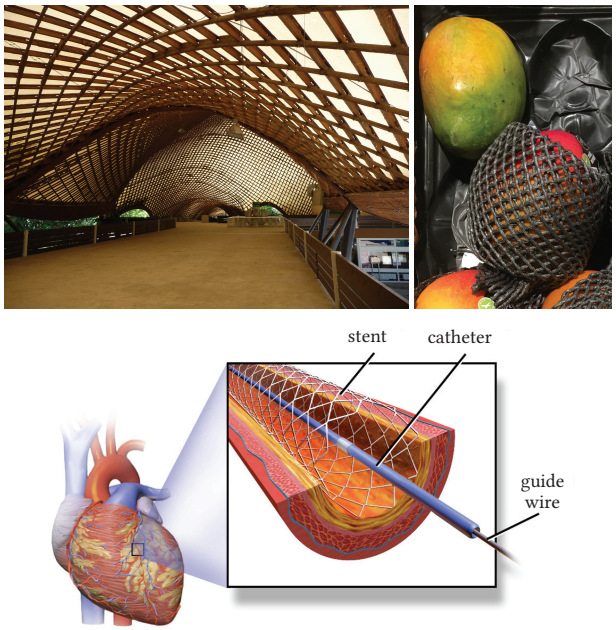


Fig. 2. Three examples of Chebyshev nets: Mannheim Multihalle architectural gridshell (top left, Wikipedia, public domain [Giel, Immanuel 2010]), fruit packaging from a supermarket (top right), a stent in a coronary artery (bottom, Wikipedia, CC BY [Blausen 2014] with title removed).

equal, and singularities are vertices with degree different than 4. Hence, our goal lies in the domain of quadrangular mesh design with automatically placed singularities, which has seen great progress in recent years. Specifically, one of the most robust and prominent approaches is to first compute a *global seamless parameterization*, and uniformly sample its iso-lines to extract a quadrangular mesh.

Parameterizing a surface can be done either through the tangents of the parameter lines, which we denote as the *guiding fields*, or through the gradients of the 2D coordinate functions. The condition for a pair of vector fields to integrate into a parameterization is different for these two formulations. Candidate gradient fields should be *curl-free*, while the guiding fields should *commute*, i.e. have vanishing Lie bracket.

To the best of our knowledge, all existing methods for seamless global parameterization use candidate gradients, and enforce a curl-free solution to guarantee that a parameterization exists. However, a Chebyshev parameterization is naturally expressed using guiding fields: a necessary and sufficient condition is that they have *unit length*. Therefore, to employ a general purpose seamless parameterization algorithm, we devise an alternative notion of *commuting guiding fields*. By additionally asking for unit length, we can approximately integrate the guiding fields into a Chebyshev net.

We present an optimization algorithm that computes a Chebyshev net covering an arbitrary input shape *automatically*. To this end, we compute a pair of vector fields on the target surface, encoded as a single PolyVector field that allows singularities. The vector fields are optimized to have unit length and to *commute*. We then integrate the resulting fields into a Chebyshev net, and extract quad meshes with near unit length edges. Finally, a gentle post-process optimization produces a discrete Chebyshev net using standard techniques.

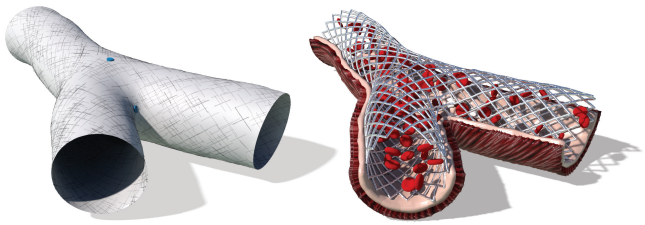


Fig. 3. Given a branched target arterial geometry (left), our algorithm automatically finds an integrable guiding field with optimal singularities that yield a global discrete Chebyshev net for a stent reinforcement (right).

We demonstrate our results on a variety of architectural and general meshes, and show Chebyshev nets that were not possible to compute previously. We additionally use our parameterization layout as a guide for cutting and fastening *real material* (Fig. 1), demonstrating that the resulting mesh is in excellent agreement with the computed quad mesh.

1.1 Contributions

Our main technical contributions are:

- We formulate the patch parameterization problem in terms of *commuting vector fields*, and formally discuss its relation to *curl-free* vector fields. We generalize this approach to global parameterizations with singularities, by using *commuting PolyVector fields*.
- We design, discretize, and solve an optimization problem for finding unit-length commuting PolyVector fields.
- We formulate and prove a relationship between Chebyshev nets and Killing vector fields on surfaces.
- We generate, for the first time, Chebyshev nets with *automatically placed singularities* for a given input triangle mesh.

2 RELATED WORK

2.1 Chebyshev nets

Chebyshev net refers either to a discrete or continuous object. *Discrete* Chebyshev nets are quad meshes with all edge lengths equal. *Continuous* Chebyshev nets are a collection of surface patches parameterized with two unit speed parameters. Physically, the edge length/unit speed constraint corresponds to inextensibility of rods in a two-dimensional network.

Continuous Chebyshev nets as parameterized patches. Russian mathematician Pafnuty Chebyshev came to his definition as a continuum model for woven fabric while asking the question of how to clothe an arbitrary surface, and explicitly found a solution for the (hemi)sphere [Ghys 2011; Tschebyscheff 1878]. Locally, a Chebyshev net exists around each point of a surface [Bakelman 1965; Bieberbach 1926]. However, arbitrary surfaces, even with disk-topology, are not guaranteed to exhibit a global singularity-free Chebyshev net; there exist global geometric obstructions that depend on a surface's distribution of Gaussian curvature [Hazzidakis 1879; Stoker

1969]. Nevertheless, surfaces of revolution that do not meet their axis exhibit singularity-free Chebyshev nets, despite having arbitrarily large total Gaussian curvature [Voss 1882]. Theoretical research on the existence of global Chebyshev nets is ongoing. Recent results include curvature bounds that guarantee existence either without singularities [Burago et al. 2007; Masson and Monasse 2017; Samelson 1991; Samelson and Dayawansa 1995] or with singularities [Masson 2017], on surfaces homeomorphic to a disk or plane. Continuous Chebyshev nets also arise in applications. For example, as a continuum model for networks of inextensible cords [Rivlin 1955, 1958], possibly with shearing [Adkins 1956; Pipkin 1984] or bending [Wang and Pipkin 1986] resistance. They have been used to rationalize the shape of buckled elastic gridshells [Baek et al. 2018].

Discrete Chebyshev nets as quad meshes. Defining discrete Chebyshev nets as quad meshes with unit edge lengths arose from studying discrete analogues of special mathematical surfaces [Bobenko and Pinkall 1996; Sauer 1970; Wunderlich 1951]. Each quad is considered as a discrete analogue of a continuous Chebyshev net patch. Similarly to quad meshes, vertices with valence not equal to four are interpreted as singularities. Applications for discrete Chebyshev nets include the design of architectural gridshells [Hennicke et al. 1974], of woven composites [Aono 1994], and of wire mesh sculptures [Garg et al. 2014].

Computational algorithms, therefore, seek discrete Chebyshev nets approximating an arbitrary target surface. Current methods can be broadly divided into two categories:

(1) **single-patch methods** compute a singularity-free discrete Chebyshev net that approximates a portion of a target surface. Some methods optimize for a Chebyshev net patch automatically from initial user-defined data [Robertson et al. 1981], found via genetic algorithms [Bouhaya et al. 2014], or by solving a related geometric problem [Hernández et al. 2013], while others use physical simulation [Bouhaya et al. 2009; Douthe et al. 2006; van West et al. 1990]. Another method allows the user to interactively grow a Chebyshev net patch, while remaining close to a target shape [Garg et al. 2014].
(2) **multi-patch methods** compute a global discrete Chebyshev net for a target surface by introducing singularities. Current methods grow the Chebyshev net from either a local patch [Aono 1994; Aono et al. 2001, 1996] or out of a singularity [Masson 2017]. Seams of tangential discontinuity and new singularities are then successively introduced to cover the entire shape. Therefore, the choice of initial local data from which to grow the net strongly influences the quality of the result. It is a priori unclear how to make this choice.

Here, we introduce an alternative approach to multi-patch methods that does not require a choice of initial local data. By optimizing a variational problem, our algorithm accounts for the geometric constraints of a Chebyshev net while optimally placing singularities. Specifically, we integrate a global Chebyshev net from a pair of commuting, unit length vector fields. Moreover, we do not introduce seams; we approximate global Chebyshev nets that are smooth except at isolated singularities. Our overall approach is completely automatic and related to existing quad meshing algorithms.

2.2 Quad meshing with integrable directional fields

Quad meshing is a rich subject [Bommes et al. 2013b] and often seeks seamless global parameterizations for a given mesh. We do not attempt to review all literature on global parameterizations [Bright et al. 2017; Chien et al. 2016; Fu and Liu 2016; Myles and Zorin 2013] and integer-grid matching [Bommes et al. 2013a, 2009; Campen et al. 2015], since it is beyond the scope of our contribution. A common paradigm is to design a directional field as candidate gradients for desired coordinate functions, and then integrate it [Vaxman et al. 2016]. Integration methods often solve Poisson equations or use periodic function representations [Fang et al. 2018; Ray et al. 2006]. Not all directional fields are the gradients of coordinates, so the problem is usually only approximately solved. There are multiple approaches to designing improved directional fields that optimize for *integrability*, in other words that the directional field is curl-free [Diamanti et al. 2015; Myles et al. 2014; Ray et al. 2006].

Our algorithm designs guiding fields for a Chebyshev net that are *not* pairs of candidate gradients. Instead, they are tangent vectors of the desired parameter lines, where the defining property of a Chebyshev net is naturally phrased: both tangent vectors should have unit length. The integrability condition translates from vanishing curl to vanishing Lie bracket, in other words that the pair of guiding vector fields *commute* (Sec. 4).

2.3 Local parameterization with commuting vector fields

Discretizations of a Lie bracket operator of two vector fields have been previously used to design discrete fields. These approaches leverage smooth formulations of the Lie bracket, e.g. as the commutator of differential operators [Azencot et al. 2013], or through its relation to the Levi-Civita covariant derivative [Azencot et al. 2015; de Goes et al. 2014]. However, all these approaches treat a pair of vector fields on a local patch. To the best of our knowledge, the generalization of a Lie bracket operator to the branched case, i.e. to PolyVector fields, has not been addressed before.

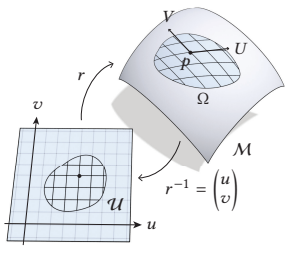
3 FORMULATION OUTLINE

Our main objective is to find a Chebyshev net on a given target surface. We first describe the main idea in the smooth setting, where the target surface is given as a compact oriented smooth surface $\mathcal{M}$ with a Riemannian metric. In Sec. 4 we restrict the exposition to *local patches* of the surface, and then generalize to a global Chebyshev net that covers the full surface in Sec. 5. In Sec. 6 we formulate the smooth optimization problem that we work with, and show how to discretize it and optimize it in Sections 7 and 8, respectively.

4 CHEBYSHEV NETS ON PATCHES

4.1 Nets as coordinate systems

In the continuous setting, a net on a surface patch $\Omega \subset \mathcal{M}$ is modeled as a *coordinate system*. That is, Ω is parameterized by two variables (u, v) in some parameter domain $\mathcal{U} \subset \mathbb{R}^2$ via a map $r\colon \mathcal{U} \to \Omega$ positioning



each point as $p = r(u, v) \in \Omega$. In reverse, u, v are also viewed as the *coordinate functions* which are the components of the inverse map $r^{-1} \colon \Omega \to \mathbb{R}^2$ that reads off the coordinates $r^{-1}(p) = (u(p), v(p))^\intercal \in \mathbb{R}^2$. The parameterization is assumed to be regular with positive orientation, namely the *coordinate vectors*

$$U \coloneqq \frac{\partial r}{\partial u}, \quad V \coloneqq \frac{\partial r}{\partial v}$$

form a basis, or *frame*, for the tangent space $T_p\mathcal{M}$

$$F_p = \begin{pmatrix} U_p & V_p \end{pmatrix}, \quad \det(F_p) > 0.$$

We write the space of frame fields over Ω as

$$\Phi_\Omega \coloneqq \{F \,|\, F_p \in \Phi_p \text{ for } p \in \Omega\} \tag{1}$$

where

$$\Phi_p \coloneqq \left\{F_p \in T_p\mathcal{M} \times T_p\mathcal{M} \,\middle|\, \det(F_p) > 0\right\} \tag{2}$$

is the set of positively oriented pairs of vectors. Note that the inverse of the frame F_p is the Jacobian of the coordinate map

$$F_p^{-1} = \begin{pmatrix} U_p & V_p \end{pmatrix}^{-1} = dr^{-1}|_p = \begin{pmatrix} du \\ dv \end{pmatrix}_p.$$

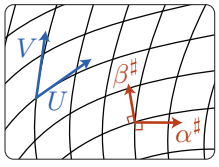
From $F_p^{-1}F_p = I$ one particularly has $dv(U) = du(V) = 0$ for all $p \in \Omega$. In other words, U is parallel to the level lines of v and V is parallel to the level lines of u. The coordinate vector fields U, V are therefore a pair of *guiding fields* for a net, where the net refers to the pattern formed by the two families of level lines of the coordinate functions u, v, one family per coordinate, sampled with equally spaced coordinate values.

A net is a *Chebyshev net* if the underlying coordinate system has unit coordinate vectors $|U| = |V| = 1$. What this means is that the level lines of v—the integral curves of U—are arclength-parameterized by u and vice versa.

4.2 Synthesizing vectors into coordinates

We reduce the problem of finding a Chebyshev coordinate system to the problem of finding a pair of guiding vector fields U, V. Working with these "differential quantities" is attractive since the defining property for a Chebyshev net only amounts to pointwise algebraic constraints $|U| = |V| = 1$ rather than differential relations. The only remaining question is the integration problem: given a pair of guiding fields U, V, how do we find a pair of coordinates u, v such that U, V are its coordinate vectors? What *integrability conditions* do U, V need to satisfy to ensure the existence of these coordinates?

Following Sec. 4.1, we define the frame field $F \coloneqq \begin{pmatrix} U & V \end{pmatrix} \in \Phi_\Omega$ from a given pair of linearly independent, positively oriented vector fields U, V over a surface patch $\Omega \subset \mathcal{M}$. The inverse of the frame produces two 1-forms α, β and corresponding vector fields $\alpha^\sharp, \beta^\sharp$ (see inset):[1]



$$F^{-1} =: \begin{pmatrix} \alpha \\ \beta \end{pmatrix}.$$

[1]The covectors (α, β) are known as the *dual basis* for the dual space T_p^*M with respect to the basis (U, V) for the tangent space at p.

The desired coordinates are then solutions to

$$du = \alpha, \quad dv = \beta. \tag{3}$$

The necessary and sufficient conditions for the local existence of (3) is that the 1-forms α, β are *closed*

$$d\alpha = d\beta = 0, \tag{4}$$

also known as the *curl-free* conditions.

4.3 Existing approaches to (3)

In the surface quadrangulation literature, there have been multiple state of the art solutions to (3) and (4). A straightforward approach to the problem could be to incorporate the unit norm constraint $|U| = |V| = 1$ into an existing scheme. Unfortunately, this is challenging with existing strategies. To clarify this point, we give a brief overview of the existing methods to the coordinate integration problem before returning to our formulation in Sec. 4.4.

Poisson Solve. Regardless of the closedness condition (4) one can solve (3) in the least squares sense

$$d \star du = d \star \alpha, \quad d \star dv = d \star \beta. \tag{5}$$

The Poisson equations (5) effectively find the potentials u, v for the exact part of α, β in their Helmholtz–Hodge decompositions. However, removing the non-exact part modifies the guiding field and one loses control of both the norms $|U|$ and $|V|$.

Curl-free fields. The work of Diamanti et al. [2015] incorporates (4) and formulates optimal designs of α, β so that they are as-closed-as-possible. The subsequent Poisson problem (5) can therefore faithfully solve (3). However, there is no clean translation of the desired Chebyshev net condition $|U| = |V| = 1$ into the design of the 1-forms α, β. Note that the magnitudes $|\alpha| \approx |du|, |\beta| \approx |dv|$ only measure the orthogonal spacing of the level lines, not the edge arclengths. Moreover, $\alpha^\sharp, \beta^\sharp$ are orthogonal to the parameter lines, whereas U, V are tangent; therefore, methods that target the design of $\alpha^\sharp, \beta^\sharp$ might not achieve the intended purpose of user alignment to constraints—unless the coordinates lines are orthogonal.

4.4 Integrability condition

We formulate the integrability conditions directly in terms of the guiding fields U, V. By evaluating the 2-forms $d\alpha, d\beta$ with the vectors U, V we have [Lee 2013, Prop. 14.29]:

$$\begin{aligned} 0 &= (d\alpha)(U, V) = d_U(\alpha(V)) - d_V(\alpha(U)) - \alpha([U, V]) \\ 0 &= (d\beta)(U, V) = d_U(\beta(V)) - d_V(\beta(U)) - \beta([U, V]), \end{aligned}$$

as α and β are closed. Moreover, since $\alpha(V) = \beta(U) = 0$ and $\alpha(U) = \beta(V) = 1$ we have

$$\begin{aligned} 0 &= (d\alpha)(U, V) = -\alpha([U, V]) \\ 0 &= (d\beta)(U, V) = -\beta([U, V]). \end{aligned}$$

Since U, V is a (non-degenerate) basis, the integrability condition (4) is equivalent to the vanishing of the *Lie bracket* between U, V

$$[U, V] = 0. \tag{6}$$

Since we are on a Riemannian manifold which comes with the torsion-free Levi-Civita connection ∇ with $\nabla_U V - \nabla_V U = [U, V]$, we also write (6) alternatively as

$$\nabla_U V - \nabla_V U = 0. \tag{7}$$

Moreover, (6) means the associated directional derivatives *commute*

$$d_U \circ d_V = d_V \circ d_U.$$

This condition can be understood intuitively as creating consistent infinitesimal quads: if we travel along a U vector and then a V vector, we reach the same point as if we had traveled the other way around. Thus there exist integrals u, v over the surface patch that record exactly the number of grids traversed along the guiding fields U, V. In particular, if $|U| = |V| = 1$, we obtain a Chebyshev net.

5 GLOBAL CHEBYSHEV NETS WITH 1/4-SINGULARITIES

So far we formulated Chebyshev nets on a surface patch $\Omega \subset \mathcal{M}$. On this sufficiently small patch a coordinate system can be described in terms of a pair of commuting vector fields. However, this representation may not be possible over a larger domain, particularly for domains that contain homologically nontrivial loops or singularities of the guiding fields. Obstructions arise from being unable to globally choose U and V.

5.1 Ambiguities in pairs of vector fields

Consider the permutation that acts on the set Φ_p (see (2)) of positively-oriented pairs of vectors

$$\Pi \colon \Phi_p \to \Phi_p, \quad \Pi(U_p, V_p) \coloneqq (-V_p, U_p),$$

which generates a cyclic group of order 4:

$$\Pi^2(U_p, V_p) = (-U_p, -V_p), \quad \Pi^3(U_p, V_p) = (V_p, -U_p), \quad \Pi^4 = \mathrm{id}.$$

The four elements $(U_p, V_p), \Pi(U_p, V_p), \Pi^2(U_p, V_p), \Pi^3(U_p, V_p)$ in Φ_p represent the same guiding field at p. Conversely, each guiding field at p has exactly these four representatives. Therefore, designing a guiding field of a net amounts to assigning to each p an element of the collection of orbits Φ_p/Π:

$$[(U, V)]_p \in \Phi_p/\Pi, \quad [(U, V)]_p = \left\{ \Pi^i(U_p, V_p) \,\middle|\, i \in \mathbb{Z}_4 \right\}.$$

In each surface patch $\Omega \subset \mathcal{M}$, the field $[(U, V)] \in \Phi_\Omega/\Pi$ has four *lifts* into pairs of vector fields $(U, V) \in \Phi_\Omega$. However, after tracking a lift continuously around a non-contractible loop (e.g. around a singularity) one may return to a different lift. The index of each singularity is an integer multiple of 1/4 since $\Pi^4 = \mathrm{id}$.

Note that despite the ambiguity, the notion of a Chebyshev net is well defined away from singularities, and invariant to the chosen lift. Explicitly, if (U, V) is integrable, i.e. $\nabla_U V = \nabla_V U$ and $|U| = |V| = 1$, then so are the other lifts of $[(U, V)]$. We can therefore formulate the Chebyshev net problem in terms of guiding fields not only on local patches Ω, but also on the entire $\mathcal{M}$ using the globally defined object $[(U, V)] \in \Phi_{\mathcal{M}}/\Pi$.

5.2 PolyVectors

The technique of PolyVectors provides an elegant way to parameterize the seemingly abstract space Φ/Π, and in general tuples of vectors modulo permutations. The technique was introduced only in the discrete setting [Diamanti et al. 2014]. For completeness, we give an exposition of the continuous formulation.

We choose a smooth orthonormal basis $b_p, Jb_p \in T_p\mathcal{M}$ at each p in some local patch $\Omega \subset \mathcal{M}$, where $|b_p| = 1$ and J is the rotation counterclockwise by 90°. Then, each vector field U on Ω is represented as a complex-valued function $\hat{U} \colon \Omega \to \mathbb{C}$, with $U_p = Re\left(\hat{U}(p)\right) b_p + Im\left(\hat{U}(p)\right) Jb_p$ for $p \in \Omega$. Equivalently, one can simply write $U = \hat{U}b$ by identifying $ib = Jb$. Likewise, a tuple of vector fields $(U_1, \ldots, U_m)$ is a $\mathbb{C}^m$-valued function $(\hat{U}_1, \ldots, \hat{U}_m)$ over Ω under the basis. To describe such a tuple of vectors modulo permutations, one views $(\hat{U}_1(p), \ldots, \hat{U}_m(p))$ at each p as the m roots of the polynomial

$$P_p(z) \coloneqq \prod_{i=1}^{m} (z - \hat{U}_i(p)) = z^m + \sum_{j=0}^{m-1} \hat{X}_j(p) z^j.$$

The coefficients $\hat{X}_0, \ldots, \hat{X}_{m-1}$ of this smoothly varying polynomial, known as a *PolyVector field*, constitute a representation of the field $(\hat{U}_1, \ldots, \hat{U}_m)$ modulo permutations.

In our case, the field $[(U, V)]$ is represented via the PolyVector

$$P(z) = (z - \hat{U})(z - \hat{V})(z + \hat{U})(z + \hat{V}) = z^4 + \hat{X}_2 z^2 + \hat{X}_0.$$

The transformation from the guiding field to its PolyVector is then

$$PV([(U, V)]) = (\hat{X}_0, \hat{X}_2), \text{ with } \hat{X}_0 = \hat{U}^2\hat{V}^2, \hat{X}_2 = -(\hat{U}^2 + \hat{V}^2). \tag{8}$$

Conversely, given $\hat{X}_0, \hat{X}_2$, one can reconstruct the guiding field $[(U, V)]$ by assigning the four roots of $P(z)$

$$\pm\sqrt{\frac{-\hat{X}_2 \pm \sqrt{\hat{X}_2^2 - 4\hat{X}_0}}{2}} \tag{9}$$

to $(\hat{U}, \hat{V}, -\hat{U}, -\hat{V})$ so that $(\hat{U}, \hat{V})$ is positively oriented. This defines PV^{-1} up to Π.

The differential calculus for the PolyVector fields involves covariant derivatives, similar to those used for vector fields. In the case of a vector field, the covariant derivative in a direction W is represented in the basis as

$$\nabla_W U = (\nabla_W \hat{U})b, \quad \nabla_W \hat{U} = (d_W - iA(W))\hat{U},$$

where A is the 1-form that measures the rotation speed of the basis field b. Alternatively, $\nabla\hat{U} = (d - iA)\hat{U}$ is a complex-valued 1-form. The corresponding covariant derivatives for $\hat{X}_0$ and $\hat{X}_2$ are given by

$$\nabla\hat{X}_0 = (d - 4iA)\hat{X}_0, \quad \nabla\hat{X}_2 = (d - 2iA)\hat{X}_2.$$

Under a change of basis $b \mapsto e^{-i\theta}b$, the basis-dependent quantities change according to

$$\begin{aligned} &\hat{U} \mapsto e^{i\theta}\hat{U}, \quad \hat{V} \mapsto e^{i\theta}\hat{V}, \quad A \mapsto A + d\theta, \\ &\hat{X}_0 \mapsto e^{4i\theta}\hat{X}_0, \quad \hat{X}_2 \mapsto e^{2i\theta}\hat{X}_2. \end{aligned} \tag{10}$$

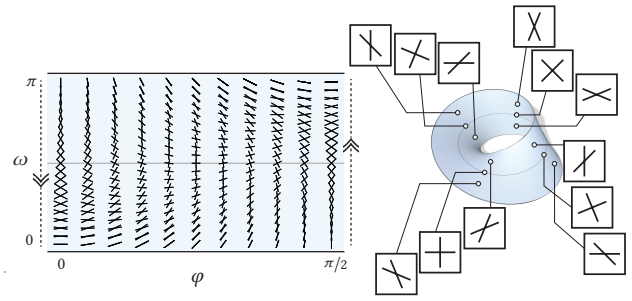


Fig. 4. The configuration space of a unit length field $[(U, V)]$ at p forms a Möbius strip. It is parameterized by the rotation angle φ of the frame $[(U, V)]_p$ from the positive real axis, and the angle ω between the frame vectors measured in positive orientation. The frame $[(U, V)]_p$ does not distinguish between ω and $\pi - \omega$. The spine (in gray) defined by $(\varphi, \pi/2)$ corresponds to orthogonal frames, and changing ω shears the frame.

These transformation rules ensure that the covariant derivatives transform like invariant objects

$$\nabla\hat{U} \mapsto e^{i\theta}\nabla\hat{U}, \quad \nabla\hat{X}_0 \mapsto e^{4i\theta}\nabla\hat{X}_0,$$
$$\nabla\hat{X}_2 \mapsto e^{2i\theta}\nabla\hat{X}_2. \tag{11}$$

5.3 Discussion: Geometry of unit length guiding fields

In Sec. 5.2 we showed that the space Φ_p/Π can be algebraically parametrized with PolyVectors $\hat{X}_0$ and $\hat{X}_2$ in $\mathbb{C}$. Here we illustrate geometrically how $\hat{X}_0, \hat{X}_2$ control the guiding field $[(U, V)]_p \in \Phi_p/\Pi$ at the point p. As Chebyshev nets are our primary interest, we only consider unit length guiding fields $|U_p| = |V_p| = 1$. It turns out that with the constraints $|U_p| = |V_p| = 1$, the topological space

$$\left(\Phi_p \cap \left\{|U_p| = |V_p| = 1\right\}\right)/\Pi \tag{12}$$

is a Möbius strip as shown in Fig. 4.

To see this, consider the guiding field $[(U, V)]_p$ at the point p, which can be visualized as a pair of lines through the origin. There exists an angular bisecting unit vector $e^{i\varphi}$, $\varphi \in [0, \frac{\pi}{2})$ in the first quadrant, such that $\hat{U}_p = e^{i\varphi}e^{-i\frac{\omega}{2}}$ and $\hat{V}_p = e^{i\varphi}e^{i\frac{\omega}{2}}$, where $\omega \in (0, \pi)$ is the angle between the lines. We can therefore parametrize a positively oriented pair of nonparallel vectors $\hat{U}_p, \hat{V}_p$ by $(\varphi, \omega) \in [0, \frac{\pi}{2}) \times (0, \pi)$. In this setup, the PolyVector coefficient $\hat{X}_0 = e^{4i\varphi}$ encodes the angular bisector, while the other PolyVector coefficient is $\hat{X}_2 = -2e^{2i\varphi}\cos\omega$. Taken together, we find the relation $\cos\omega = \pm\frac{\hat{X}_2}{2\sqrt{\hat{X}_0}}$. The sign ambiguity in solving for $\cos\omega$ introduces an identification of $(0, \omega)$ along $(\frac{\pi}{2}, \pi - \omega)$ in (φ, ω)-parameter space, which yields a Möbius strip. Roughly speaking, the PolyVector coefficients cannot distinguish pairs of positively-oriented lines with an oriented angle ω from those with an oriented angle $\pi - \omega$.

6 OPTIMIZATION PROBLEM

Our goal is to find a Chebyshev net on $\mathcal{M}$ that is as-smooth-as-possible away from an optimal set of isolated 1/4-singularities. We synthesize this Chebyshev net from a continuous guiding field $[(U, V)] \in \Phi_{\mathcal{M}}/\Pi$, which is found by solving a variational problem that automatically introduces singularities in locations adapted to the geometric constraints.

6.1 Geometric constraints

Away from its singularities, the guiding field $[(U, V)]$ should satisfy the following conditions:

(1) **Integrability.** The vector fields commute:

$$|\nabla_U V - \nabla_V U|^2 = 0.$$

(2) **Unit length.** The vector fields are unit length:

$$|U|^2 = |V|^2 = 1.$$

(3) **Angle bound.** The angles are bounded by $\zeta > 0$:

$$\min_{(U,V)\in[(U,V)]} \langle U, V\rangle \geq \cos\zeta.$$

These three conditions are invariant to an explicit choice of lift to (U, V), $(-V, U)$, $(-U, -V)$, or $(V, -U)$. Therefore, they are well defined on $[(U, V)]$.

6.2 Ginzburg–Landau energy

At a singularity, the guiding field $[(U, V)]$ has $U = V = 0$, so the unit-length constraint cannot be satisfied. Instead of relaxing the constraint [Knöppel et al. 2013], we follow the methodology of Viertel and Osting [2019] and formulate the unit-length constraint using a Ginzburg–Landau approach. We consider the energy

$$E_{\mathrm{GL}}([(U, V)]) = E_{\mathrm{int}}([(U, V)]) + \frac{1}{\varepsilon^2}E_{\mathrm{norm}}([(U, V)]), \tag{13}$$

where

$$E_{\mathrm{int}}([(U, V)]) = \int_{\mathcal{M}} |\nabla_U V - \nabla_V U|^2 \text{ and} \tag{14}$$
$$E_{\mathrm{norm}}([(U, V)]) = \int_{\mathcal{M}} \left((|U|^2 - 1)^2 + (|V|^2 - 1)^2\right). \tag{15}$$

Minimizing E_{GL} is similar to minimizing the deviation from integrability subject to the unit-length constraint. This is analogous to the following well known fact: as $\varepsilon \to 0$, minimizing the *Ginzburg–Landau energy*

$$\int_{\mathcal{M}} |\nabla U|^2 + \frac{1}{\varepsilon^2}(|U|^2 - 1)^2$$

is asymptotically equivalent to minimizing the Dirichlet energy of a unit-length vector field over the manifold $\mathcal{M}$ after removing a ball of radius ε around each singularity. The energy of minimizers of the Ginzburg–Landau energy is bounded away from zero in the presence of singularities [Bethuel et al. 1994].

A significant advantage of a Ginzburg–Landau approach to satisfying the integrability and unit-length constraints on $[(U, V)]$ is that singularities of the guiding field will automatically arise during optimization. Fig. 5 demonstrates the importance of allowing the singularities to move freely.

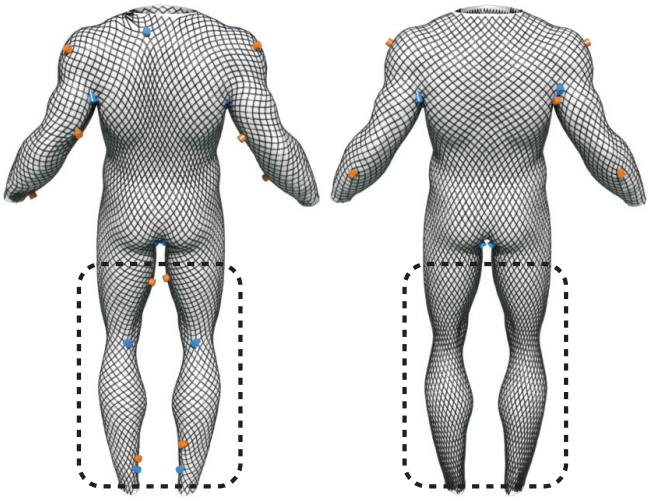


Fig. 5. Comparing between an approximate Chebyshev net computed using our algorithm, with prescribed fixed singularities (left) vs automatically arising from our Ginzburg–Landau optimization (right). The prescribed singularities are optimal for orthogonal cross fields, and are found using the algorithm of [Knöppel et al. 2013]. Our optimization yields singularities that are adapted to unit length and integrable guiding fields. In this example, our optimization removes the unnecessary singularities on the legs leading to a more evenly distributed shearing of the Chebyshev net.

6.3 As-smooth-as-possible guiding fields

We regularize the guiding fields $[(U, V)]$ using the Dirichlet energies arising from the explicit PolyVector parameterization:

$$E_{\text{reg}}(\hat{X}_0, \hat{X}_2) = \int_{\mathcal{M}} |\nabla \hat{X}_0|^2 + |\nabla \hat{X}_2|^2, \tag{16}$$

where the differential operators are defined for $\hat{X}_0$ and $\hat{X}_2$ in Sec. 5.2.

Minimizing this energy alongside E_{GL} leads to optimal singularity locations that achieve the tradeoff between smoothness of the unit-length guiding fields and their integrability.

6.4 Continuous optimization problem

We solve the following optimization problem for continuous guiding fields $[(U, V)]$ parameterized by complex-valued PolyVectors $\hat{X}_0, \hat{X}_2$ that lead to Chebyshev nets with optimally placed singularities:

$$\begin{aligned} &\underset{[(\hat{U},\hat{V})]}{\text{minimize}} \quad \lambda_{\text{GL}} E_{\text{GL}}([(\hat{U}, \hat{V})]) + \lambda_{\text{reg}} E_{\text{reg}}(\hat{X}_0, \hat{X}_2) \\ &\text{subject to } \hat{X}_0 = \hat{U}^2 \hat{V}^2, \\ &\qquad \hat{X}_2 = -\left(\hat{U}^2 + \hat{V}^2\right), \text{ and} \\ &\qquad \min_{(\hat{U},\hat{V}) \in [(\hat{U},\hat{V})]} \langle \hat{U}, \hat{V} \rangle \geq \cos \zeta, \end{aligned} \tag{17}$$

for scalar parameters $\lambda_{\text{GL}}, \lambda_{\text{reg}} > 0$. Specifically, we seek solutions in the limit $\lambda_{\text{reg}}/\lambda_{\text{GL}} \to 0$ so that, away from their optimally placed singularities, the resulting fields will be as-integrable-as-possible.

The angle bound constraint prevents the trivial solution $\hat{U} = \hat{V}$. However, enforcing this constraint requires working with the lifts $\hat{U}$ and $\hat{V}$ explicitly. This is because PolyVector coefficients admit the ω and $\pi - \omega$ ambiguity discussed in Sec. 5.3.

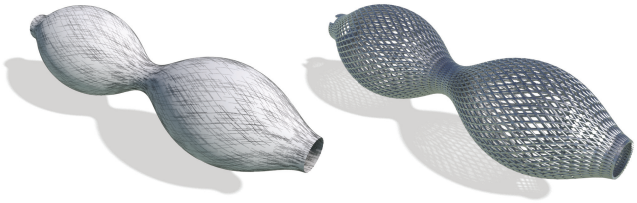


Fig. 6. On a surface of revolution, our initial Killing vector field solution leads to a guiding field (left) that is unit length and commuting, so it integrates into a Chebyshev net (right) without any further integrability optimization.

6.5 Initialization from approximate Killing vector fields

The optimization problem we consider is nonconvex and benefits from good initialization. Our initial solution builds from the following theorem, which is a straightforward calculation (given in Appendix A), but we believe has not been formulated before. As a reminder, Killing vector fields are ones that generate an intrinsic isometric flow [Lee 2013, pp 343].

Theorem 1. *Let $\mathcal{M}$ be a 2-dimensional Riemannian manifold that admits a non-vanishing Killing vector field Y with $|Y| < 1/c_0$ for some global constant $c_0 \in \mathbb{R}$. Then for each $0 < c < c_0$ the vector fields*

$$\begin{aligned} &U := cY + s(|Y|)JY \quad \text{and} \quad V := cY - s(|Y|)JY \\ &\text{with } s(t) := \sqrt{t^{-2} - c^2} \end{aligned} \tag{18}$$

are linearly independent and satisfy $|U| = |V| = 1$ and $[U, V] = 0$. In particular, $\mathcal{M}$ admits a global Chebyshev net that is singularity-free.

Remark 1. *This generalizes a result by A. Voss [1882], which states that surfaces of revolution that do not meet their axis exhibit global Chebyshev nets that are singularity-free. Surfaces of revolution are representative examples of surfaces that admit Killing vector fields (Fig. 6).*

We therefore initialize our optimization by computing an *approximate Killing vector field* Y [Azencot et al. 2013; Ben-Chen et al. 2010], and then define our initial solution to be the U, V given by Equation (18) with $c = c_0/5$. In the special case that the surface is exactly a surface of revolution, our initial solution is already a Chebyshev net (see Fig. 6). On a general surface, the initial solution provides a good initialization for our optimization scheme.

In what follows, we show how to discretize and solve this optimization problem in practice.

7 DISCRETIZATION

Our computational setup takes an input target surface as a triangle mesh. We next formulate an optimization problem approximating (17) using guiding fields discretized on the input mesh.

7.1 Representation

Notation. The surface is represented as an orientable triangle mesh $\mathcal{M} = \{\mathcal{V}, \mathcal{E}, \mathcal{F}\}$ with arbitrary genus and boundary components. The edge vector of the mutual edge of two faces $f_i, f_j \in \mathcal{F}$ is

denoted by e_{ij}. In the basis of face f_j, we represent the dual edge e^*_{ij} as $\hat{e}^*_{ij} = i\frac{h_{ij}}{|\hat{e}_{ij}|}\hat{e}_{ij}$, where h_{ij} is the sum of the barycenter heights in both faces with respect to the mutual edge.

Vector fields. We denote by $\mathcal{X}$ the space of face-based piecewise-constant vector fields. We construct the complex representation for $\mathcal{X}$ by defining a local (arbitrary) orthonormal basis b_f, Jb_f per face $f \in \mathcal{F}$. The representation of a tangent vector $V_f \in T_f\mathcal{M}$ at the face f is given by the complex number $\hat{V}_f = \langle V_f, b_f\rangle + i\langle V_f, Jb_f\rangle$ [Knöppel et al. 2013]. Hence, $\hat{V} \in \mathbb{C}^{|\mathcal{F}|}$.

Connections. The connection 1-form A is discretized by a complex number r_{ij} on dual edges e^*_{ij}, where r_{ij} is a discontinuous "jump" across the edge. Explicitly, $r_{ij} = \hat{e}_j/\hat{e}_i$, where $\hat{e}_i$ (resp. $\hat{e}_j$) is the complex number representing the mutual edge vector e_{ij} in the basis of face f_i (resp. f_j). The discrete covariant derivative of a vector field $\hat{V}$ in the direction of a dual edge e^*_{ij} is given by

$$D_{|ij}\hat{V} = \hat{V}_i r_{ij} - \hat{V}_j, \tag{19}$$

and is with respect to the basis at f_j. It can be therefore represented by a sparse complex matrix $D \in \mathbb{C}^{|\mathcal{E}|\times|\mathcal{F}|}$.

PolyVectors. The PolyVector coefficients $\hat{X}_0, \hat{X}_2$ are similarly represented as face-based complex numbers [Diamanti et al. 2014], where $\hat{X}_{0,i}$ represents $\hat{X}_0$ in the face f_i using the corresponding local basis. The discrete covariant derivative of the PolyVector coefficients in the direction of the dual edge e^*_{ij} is given by:

$$D^{(0)}_{|ij}\hat{X}_0 = \hat{X}_{0,i}r^4_{ij} - \hat{X}_{0,j}, \quad D^{(2)}_{|ij}\hat{X}_2 = \hat{X}_{2,i}r^2_{ij} - \hat{X}_{2,j}, \tag{20}$$

with respect to the basis at f_j. These can similarly represented as sparse complex matrices $D^{(0)}, D^{(2)} \in \mathbb{C}^{|\mathcal{E}|\times|\mathcal{F}|}$.

Lifts and matchings. The transformation PV^{-1} that computes $(\hat{U}_f, \hat{V}_f)$ from the PolyVector coefficients $(\hat{X}_{0,f}, \hat{X}_{2,f})$ at every face $f \in \mathcal{F}$ is defined only up to Π. When computing the covariant derivative of one of the lifted vector fields, we need to take into consideration that neighboring triangles may have different lifts. We therefore define an edge-based permutation matrix, parameterized by an integer κ_{ij}, given by $\pi_{ij} = \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)^{\kappa_{ij}}$, and incorporate it into the definition of the covariant derivative, in the direction of a dual edge, of a lift $(\hat{U}, \hat{V})$ as follows:

$$\mathcal{D}_{|ij}\begin{pmatrix}\hat{U}\\ \hat{V}\end{pmatrix} = \pi_{ij}\begin{pmatrix}\hat{U}_i\\ \hat{V}_i\end{pmatrix} r_{ij} - \begin{pmatrix}\hat{U}_j\\ \hat{V}_j.\end{pmatrix} \tag{21}$$

The covariant derivative $\mathcal{D}_{|ij}(\hat{U}, \hat{V})$ is locally a complex vector with two entries, represented in the basis of face f_j.

Singularities. The integer κ_{ij} is called the (order-preserving) *matching* between the vectors across an edge. The fractional index of a vertex $v \in \mathcal{V}$ is defined as

$$\kappa_v = \tfrac{1}{4}\mathrm{mod}\big(\textstyle\sum_{e_{ij}\in N(v)}\kappa_{ij}, 4\big),$$

where a vertex is *singular* if $\kappa_v \neq 0$.

Principal matching. In general, π_{ij} can be chosen arbitrarily, which will affect the covariant derivatives and the singularity structure. We choose a specific matching, denoted as the *principal matching*, which is the matching that leads locally to a smooth choice of lift. Specifically, we do a closest-angle matching on the four rotationally-symmetric roots of $\hat{X}_0$, which are the bisectors between the $(\hat{U}, \hat{V})$ vectors, and consequently match the vectors $(\hat{U}, \hat{V})$ according to the resulting matching sectors.

7.2 Discretized integrability energy

To formulate the integrability energy in the discrete case we require a discretization of $\nabla_U\hat{V}$, whereas Equation (19) is a discretization of only $\nabla_{e^*_{ij}}\hat{V}$. To that end, we represent $\nabla\hat{V}$ in a local orthogonal basis (Z, JZ), and note that we have:

$$\nabla_U\hat{V} = \langle U, Z\rangle\nabla_Z\hat{V} + \langle U, JZ\rangle\nabla_{JZ}\hat{V}. \tag{22}$$

Taking $Z = e_{ij}/|e_{ij}|$, and noting that the covariant derivative parallel to the edge e_{ij} is 0 for piecewise-constant fields, leads to:

$$(D_U\hat{V})_{|ij} = \frac{1}{|e^*_{ij}|^2}\langle U, e^*_{ij}\rangle D_{|ij}\hat{V}. \tag{23}$$

This covariant derivative is further generalized to a branched $(\hat{U}, \hat{V})$ field by employing (21) with the principal matching:

$$\left(\mathcal{D}_U\begin{pmatrix}\hat{U}\\ \hat{V}\end{pmatrix}\right)_{|ij} = \frac{1}{|e^*_{ij}|^2}\langle U, e^*_{ij}\rangle\mathcal{D}_{|ij}\begin{pmatrix}\hat{U}\\ \hat{V}\end{pmatrix}. \tag{24}$$

This leads to matrices $\mathcal{D}_U$ and similarly $\mathcal{D}_V$ of type $\mathbb{C}^{2|\mathcal{E}|\times 2|\mathcal{F}|}$ that implement the branched directional derivative.

Both (23) and (24) require evaluating $\langle U, e^*_{ij}\rangle$ on an edge. For this evaluation, we use $\hat{U}_{ij} = \frac{1}{2}(\hat{U}_i r_{ij} + \hat{U}_j)$, which is an average of $\hat{U}_i$ and $\hat{U}_j$ represented in the basis of f_j. In the branched case,

$$\begin{pmatrix}\hat{U}_{ij}\\ \hat{V}_{ij}\end{pmatrix} = \frac{1}{2}\left(\pi_{ij}\begin{pmatrix}\hat{U}_i\\ \hat{V}_i\end{pmatrix} r_{ij} + \begin{pmatrix}\hat{U}_j\\ \hat{V}_j\end{pmatrix}\right).$$

Lie Derivative. Using the discrete covariant derivatives defined above, the Lie derivative $[U, V] = \nabla_U V - \nabla_V U$ is discretized as a vector per edge ij as a complex number in the basis at face f_j:

$$\left(\mathcal{L}\begin{pmatrix}\hat{U}\\ \hat{V}\end{pmatrix}\right)_{|ij} = \begin{pmatrix}0 & 1\end{pmatrix}\left(\mathcal{D}_U\begin{pmatrix}\hat{U}\\ \hat{V}\end{pmatrix}\right)_{|ij} - \begin{pmatrix}1 & 0\end{pmatrix}\left(\mathcal{D}_V\begin{pmatrix}\hat{U}\\ \hat{V}\end{pmatrix}\right)_{|ij}$$

This defines a sparse matrix $\mathcal{L} \in \mathbb{C}^{|\mathcal{E}|\times 2|\mathcal{F}|}$ that can be constructed using the previously defined matrices:

$$\mathcal{L}\begin{pmatrix}\hat{U}\\ \hat{V}\end{pmatrix} = \begin{pmatrix}0 & I & -I & 0\end{pmatrix}\begin{pmatrix}\mathcal{D}_U\\ \mathcal{D}_V\end{pmatrix}\begin{pmatrix}\hat{U}\\ \hat{V}\end{pmatrix},$$

where $0, I$ are the zero and identity matrices of size $|\mathcal{E}|$.

Integrability energy. The integrability energy (14) is then discretized as

$$E_{\mathrm{int}}(U, V) = \begin{pmatrix}\overline{\hat{U}} & \overline{\hat{V}}\end{pmatrix}\underbrace{\overline{\mathcal{L}}^{\mathsf{T}}M_{\mathcal{E}}\mathcal{L}}_{=:B_{(U,V)}}\begin{pmatrix}\hat{U}\\ \hat{V}\end{pmatrix} \tag{25}$$

with a diagonal matrix $M_{\mathcal{E}}$ of edge areas $M_{\mathcal{E}}(e, e) = \frac{1}{2}|e||e^*|$, where $\overline{(\cdot)}$ is complex conjugation. Note that (25) is not an actual quadratic form on U and V, as $B_{(U,V)}$ depends on U and V.

7.3 The full system

The discrete version of the system (17) is given as follows:

$$\begin{aligned}
&\underset{(\hat{X}_0,\hat{X}_2,\hat{U},\hat{V})}{\text{minimize}}\ \lambda_{\text{GL}}\left[E_{\text{int}}+\frac{1}{\varepsilon^2}\left(\begin{pmatrix}|\hat{U}|^2-1\\|\hat{V}|^2-1\end{pmatrix}^{\mathsf{T}}\begin{pmatrix}M_{\mathcal{F}} & \\ & M_{\mathcal{F}}\end{pmatrix}\begin{pmatrix}|\hat{U}|^2-1\\|\hat{V}|^2-1\end{pmatrix}\right)\right]\\
&\qquad\qquad+\lambda_{\text{reg}}\left[\overline{\hat{X}_0}^{\mathsf{T}}L_0\hat{X}_0+\overline{\hat{X}_2}^{\mathsf{T}}L_2\hat{X}_2\right]\\
&\text{subject to } \hat{X}_0=\hat{U}^2\hat{V}^2,\ \hat{X}_2=-\left(\hat{U}^2+\hat{V}^2\right),\ \text{and}\\
&\qquad\qquad\langle\hat{U},\hat{V}\rangle\geq\cos\zeta,
\end{aligned}\tag{26}$$

where

- $M_{\mathcal{F}}$ is the diagonal face-based areas mass matrix.
- $L_0=\overline{D^{(0)}}^{\mathsf{T}}WD^{(0)}$ and $L_2=\overline{D^{(2)}}^{\mathsf{T}}WD^{(2)}$ are the Laplacians implementing the Dirichlet energy for the PolyVector coefficients, where $W=\text{diag}(|e|/|e^*|)_{e\in\mathcal{E}}$.
- ζ is a user-chosen angle bound. We use $\pi/6$ for all examples.

8 OPTIMIZATION

We illustrate our pipeline in Figure 7.

8.1 Initial solution

We initialize our algorithm by first finding an approximate Killing vector field Y on the surface using [Azencot et al. 2013] and then defining $\hat{U},\hat{V}$ as explained in Sec. 6.5.

8.2 Computing the guiding fields

The objective function of the optimization problem (26) has a natural separation between the variable sets $(\hat{X}_0,\hat{X}_2)$ and $(\hat{U},\hat{V})$. In addition, our constraints (angle bounds and compatibility between the two sets) can be enforced locally per face or edge. Consequently, we design an alternating optimization scheme.

We split the two variable sets, at each iteration keeping either $(\hat{X}_0,\hat{X}_2)$ or $(\hat{U},\hat{V})$ fixed, while optimizing for the other variable pair. We denote the value of the variables at iteration i with a superscript, e.g., $(\hat{U},\hat{V})^i$. We denote the PolyVector transformation as PV (Section 5.2), and the root extractions and principal matching as PV^{-1} accordingly. The algorithm is given in Algorithm 1.

8.2.1 Schedule: parameter scheduling. For all our examples, we run the algorithm for $k=60$ iterations. We keep $\lambda_{\text{GL}}=1$ constant throughout, and halve λ_{reg} every ten iterations starting from $i=20$, to let the regularization energy be more dominant in the beginning, and focus on GL integrability once the singularities have settled.

8.2.2 Reg: regularization. This step aims to improve the regularization energy E_{reg} given in Equation (16). Following [Viertel and Osting 2019], we use an implicit smoothing step to reduce the Dirichlet energy. Hence, we solve the following two linear equations:

$$\begin{aligned}
\left(M_{\mathcal{F}}+\tau_0\,\lambda_{\text{reg}}^i\,L_0\right)\hat{X}_0'&=M_{\mathcal{F}}\hat{X}_0\\
\left(M_{\mathcal{F}}+\frac{1}{2}\tau_2\,\lambda_{\text{reg}}^i\,L_2\right)\hat{X}_2'&=M_{\mathcal{F}}\hat{X}_2,
\end{aligned}\tag{27}$$

and then set $(\hat{X}_0,\hat{X}_2)\leftarrow(\hat{X}_0',\hat{X}_2')$. Here, L_2 and L_0 are the 2-monomial and 0-monomial Laplacians (Section 7.3), and $M_{\mathcal{F}}$ is the triangle-area diagonal matrix. For the time scales we take $\tau_0=\mu_0^{-1},\tau_2=\mu_2^{-1}$, where μ_0,μ_2 are the smallest non-zero eigenvalues of L_0,L_2, respectively. We use an extra factor of $1/2$ to smoothen $\hat{X}_2$ because it empirically corresponds better with subsequent integrability steps. Note that as $\hat{X}_0$ changes, so may the singularities of the field.

8.2.3 AngleUnit: projection. To adhere to the angle bound constraints, we project $(\hat{U},\hat{V})$ to the feasible set by rotating them away from the bisector symmetrically, as described in [Diamanti et al. 2014]. We note that the bisector itself (which is encoded by the resulting $\hat{X}_0$) is invariant to this operation, and thus the singularities of the entire field do not change in this step. We additionally normalize $(\hat{U},\hat{V})$ to have unit length.

8.2.4 GL: Ginzburg–Landau integrability. The goal of this step is to improve the integrability energy given in Equation (25). Noting that $B_{(U,V)}$ is positive semi-definite for any $\hat{U},\hat{V}$, we employ semi-implicit smoothing, similarly to the regularization step. We solve:

$$\left(\begin{pmatrix}M_{\mathcal{F}} & \\ & M_{\mathcal{F}}\end{pmatrix}+\lambda_{GL}B_{(U,V)}\right)\begin{pmatrix}\hat{U}'\\\hat{V}'\end{pmatrix}=\begin{pmatrix}M_{\mathcal{F}}\hat{U}\\M_{\mathcal{F}}\hat{V}\end{pmatrix},\tag{28}$$

and then set $(\hat{U},\hat{V})=(\hat{U}',\hat{V}')$.

8.2.5 Extension: diagonal alignment. We can align a Chebyshev net with a mesh's sharp features or boundaries. Contrary to the common case, our desired alignment is *diagonal*—the sharp features should align with the diagonals of quads rather than their edges, so the net folds flexibly across these features. To achieve this, we define an edge e to be a feature when its dihedral angle is bigger than $\pi/3$, or if it is a boundary edge, and denote by $C\subset\mathcal{F}$ the set of faces neighboring such edges. For every face $c\in C$ adjacent to a feature edge e_c, we constrain $\hat{X}_{0,c}$ to be $(\hat{e}_c)^4$ (the 4^{th} power representative of the unit-length vector of e_c in the basis of c), as $\hat{X}_{c,0}$ is exactly the bisector to $\hat{U}_c,\hat{V}_c$. See Figure 8 for an example. Diagonal alignment is turned on by default for all meshes shown and we did not observe problems between initialization and diagonal alignment.

Algorithm 1 Optimize Chebyshev Field

Input: mesh $\mathcal{M}=\{\mathcal{V},\mathcal{E},\mathcal{F}\}$.
Output: Chebyshev guiding fields $(\hat{U},\hat{V})$.
Compute an approximate Killing field Y. ▷ Section 8.1
Extract $(\hat{U},\hat{V})^0$ from Y. ▷ Equation (18)
$\lambda_{\text{reg}}^0:=1,\ \lambda_{\text{GL}}:=1$.
for $i=1,2,\dots k$ **do**
  $\lambda_{\text{reg}}^i\leftarrow\text{Schedule}(\lambda_{\text{reg}}^{i-1})$ ▷ Section 8.2.1
  $(\hat{X}_0,\hat{X}_2)^i\leftarrow PV\left((\hat{U},\hat{V})^{i-1}\right)$ ▷ Equation (8)
  $\forall c\in C,\hat{X}_{0,c}^i\leftarrow(\hat{e}_c)^4$ ▷ Section 8.2.5
  $(\hat{X}_0,\hat{X}_2)^i\leftarrow\text{Reg}\left((\hat{X}_0,\hat{X}_2)^i\right)$ ▷ Section 8.2.2
  $(\hat{U},\hat{V})^i\leftarrow PV^{-1}\left((\hat{X}_0,\hat{X}_2)^i\right)$ ▷ Equation (9), Section 7.1
  $(\hat{U},\hat{V})^i\leftarrow\text{AngleUnit}\left((\hat{U},\hat{V})^i\right)$ ▷ Section 8.2.3
  $(\hat{U},\hat{V})^i\leftarrow\text{GL}\left((\hat{U},\hat{V})^i\right)$ ▷ Section 8.2.4
end for

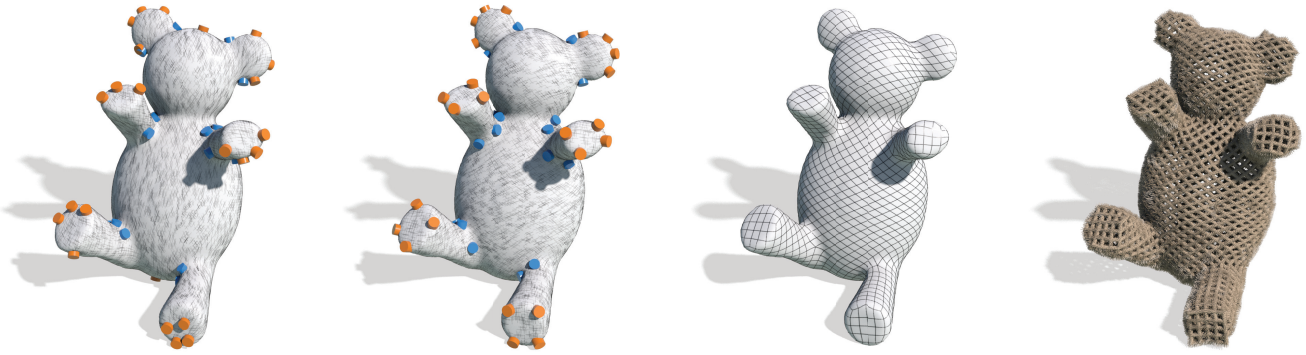


Fig. 7. Algorithm pipeline. Initial guiding fields are constructed from an approximate Killing vector field (left). Our main Algorithm 1 solves for unit length commuting guiding fields with optimally placed singularities (middle-left). The resulting guiding fields are integrated into a globally seamless parameterization, yielding a quad mesh approximating a Chebyshev net (middle-right). A ShapeUp post-processing step gives a discrete Chebyshev net (right).

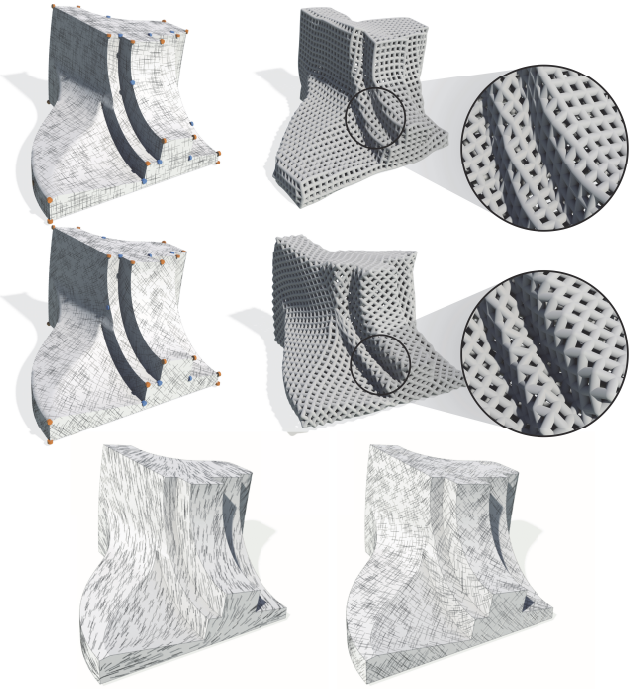


Fig. 8. Fandisk without (top row) and with (middle row) the sharp-feature alignment constraint. Our diagonal alignment algorithm allows the Chebyshev net to naturally bend over sharp features, even though the initial approximate Killing field (bottom row, left) may be far from the aligned optimal field (bottom row, right).

8.3 Computing the candidate gradient vector fields

Our next step is to extract (u, v) coordinate functions that correspond to guiding fields (U, V). To avoid explicit tracing, we use a seamless parameterization approach. Recall that our formulation computes a guiding field tangent to the integral lines (Sec. 4), so we first compute the candidate gradient vector fields $(\alpha^\sharp, \beta^\sharp)$ of the coordinate functions. The relation between the two pairs of vector fields is pointwise, and in the smooth case is given by:

$$\begin{pmatrix} \alpha_p \\ \beta_p \end{pmatrix} \begin{pmatrix} U_p & V_p \end{pmatrix} = Id.$$

The following conditions should then hold for $(\alpha_f^\sharp, \beta_f^\sharp)$ in face $f \in \mathcal{F}$:

$$\begin{aligned} \langle \alpha_f^\sharp, U_f \rangle &= 1, \quad \langle \alpha_f^\sharp, V_f \rangle = 0 \\ \langle \beta^\sharp, U_f \rangle &= 0, \quad \langle \beta_f^\sharp, V_f \rangle = 1. \end{aligned} \tag{29}$$

Since we enforce the angle bound constraints, (U_f, V_f) cannot be parallel, and therefore these linear constraints exactly determine $(\alpha_f^\sharp, \beta_f^\sharp)$ per face, up to the lift Π.

8.4 Candidate gradient vector fields to discrete Chebyshev nets

Having computed the candidate gradient vector fields, we proceed to create a Chebyshev global seamless parameterization, and consequently a discrete Chebyshev quad net.

Global parameterization. The candidate gradient vector fields $(\alpha_f^\sharp, \beta_f^\sharp)$ are defined up to some choice of lift Π per face. Following standard quadrangular remeshing schemes [Bommes et al. 2013b], we first cut the mesh such that the singularities are on the boundaries of the cut mesh, and then compute two separate vector fields. Finally, a Poisson equation is solved, taking into account the rotation and translation period jumps across cuts. A global scaling factor $1/\ell$ is used in the Poisson solve to control the density of the resulting quadrangulation; we use $\ell = 0.02$ for all our examples, unless stated otherwise. To obtain pure quadrilateral meshes, we set singularities to be in integral positions on the uv grid, using greedy rounding.

Quad-net optimization. We proceed by extracting a quad mesh from the seamless parameterization. As the Chebyshev net conditions are discretized and local, the quad mesh is only approximately a Chebyshev quad mesh. That is because the resulting quads are not infinitesimal, and therefore we get a first-order approximation at best. However, this still gives a good candidate net for post-process

optimization. We do so by employing ShapeUp [Bouaziz et al. 2012], constraining all edge lengths to be equal. Since we expect our initial quad mesh to be close enough to a Chebyshev quad mesh, we do not employ any auxiliary proximity terms to the original mesh. We use 200 iterations for all the examples we show in this paper.

9 ANALYSIS

9.1 Implementation details

We implemented our algorithm using MATLAB, where we used auxiliary C++ code from the software package Directional [Vaxman et al. 2017] to compute the seamless parameterization and trace the visualized streamlines, libQEx [Ebke et al. 2013] to extract the quad mesh from the parameterization, and libigl [Jacobson et al. 2018] to perform the ShapeUp optimization.

9.2 Timings and convergence

Our code runs on a 2.3 GHz i7 Macbook Pro with 16 GB of memory. In Table 1 we list the run times for the different steps of the algorithm, together with position error η_{pos} (30), deformation error η_{def} (31), and edge length deviation error η_{len} (32). It is evident that the guiding field optimization and initialization times are primarily correlated with the number of triangles in the input mesh, whereas the parameterization time is naturally correlated with the number of singularities. ShapeUp mesh optimization time is overall negligible.

In Figures 9 and 10 we demonstrate that our optimization effectively minimizes the integrability energy E_{int} (25) of the guiding field. As explained in Sec. 6.2, we generically do not expect convergence to zero energy. We observed that 60 iterations are sufficient for our examples to converge to a constant energy level. The resulting quad mesh is nearly a discrete Chebyshev net, so the ShapeUp optimized net remains close to the desired target shape.

9.3 Effects of integrability and post-processing

We demonstrate the necessity to optimize for integrability of the guiding field to solve for Chebyshev nets. Figure 11 shows that Poisson-integrated quadrangulations from a non-integrable guiding field may not be aligned with the underlying field. In Figure 12, a unit length guiding field results in a well-aligned Poisson-integrated quadrangulation, yet it is far from being a Chebyshev net since the guiding field is non-integrable. This causes the ShapeUp optimization to drastically alter the shape of the net. As such, maintaining the integrability of the guiding field is essential to the success of our algorithm, as quantified by the following three error measurements.

The relative position error $\eta_{\text{pos}} > 0$ is defined by:

$$\eta_{\text{pos}}^2 = \frac{1}{|\mathcal{V}_Q| \cdot R^2} \sum_{v \in \mathcal{V}_Q} |v_p - v_s|^2, \tag{30}$$

where $\mathcal{V}_Q$ is the set of vertices of the resulting quad mesh, v_p are the vertex coordinates on the target surface after parameterization, and v_s the quad mesh vertex coordinates after the ShapeUp optimization. To make η_{pos} scale invariant, R is a circumsphere of the target surface given by the longest distance between the vertices' barycenter and every vertex in v_p.

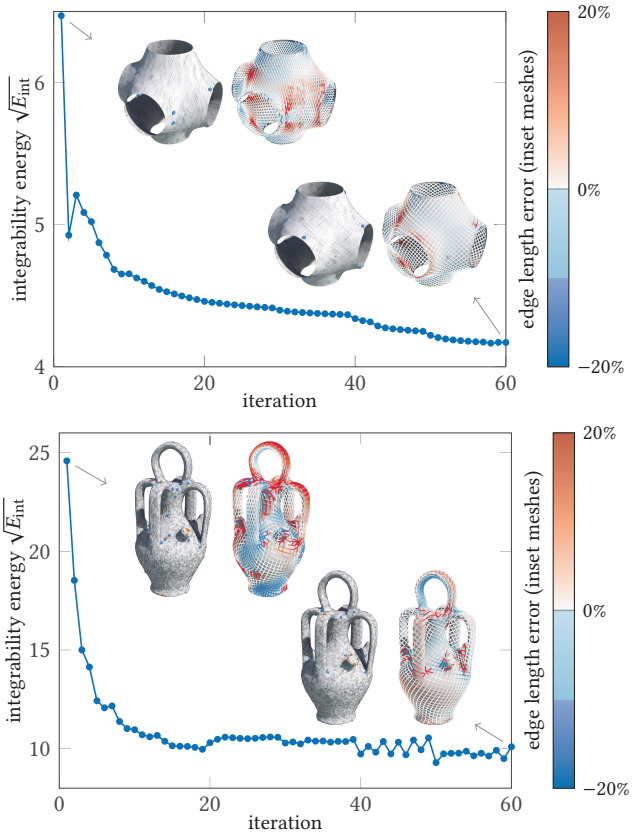


Fig. 9. The integrability energy $\sqrt{E_{\text{int}}}$ at each iteration of Algorithm 1, where the final result is shown in Figure 10. The inset models compare between the initial solution and our optimized solution over the iterations, by showing the guiding fields with singularities and the deviations from having constant edge lengths in the quad mesh obtained after parameterization, but before the ShapeUp optimization. Note that we expect E_{int} to converge to a nonzero constant (Sec. 6.2). The edge length error measures deviation from the mean.

The deformation error $\eta_{\text{def}} > 0$ is defined by:

$$\eta_{\text{def}}^2 = \frac{1}{6|Q|} \sum_{q \in Q} \sum_{(i,j) \in q} \left(\frac{|v_{i,s} - v_{j,s}|^2}{|v_{i,p} - v_{j,p}|^2} - 1 \right)^2, \tag{31}$$

where Q is the set of quads and (i, j) is every edge or diagonal in a quad $q \in Q$. The deformation error is 0 when two quad meshes are isometric while treating each quad as a tetrahedron; in other words, corresponding edge and diagonal lengths of both meshes agree.

The edge length deviation error $\eta_{\text{len}} \geq 0$ is defined by:

$$\eta_{\text{len}} = \max_{(i,j) \in \mathcal{E}_Q} \left| \frac{|v_{i,s} - v_{j,s}|}{\ell_Q} - 1 \right|, \tag{32}$$

where (i, j) is an edge from the set of quad-mesh edges $\mathcal{E}_Q$, v_s are the quad-mesh vertex coordinates after ShapeUp optimization, and ℓ_Q is the average edge length over the entire mesh. The edge length deviation error is 0 when the post-processed quad mesh has all edge lengths equal, i.e., when the final result is a discrete Chebyshev net.

Mesh	# Tri.	# Quad.	# Sing.	Run time (in seconds)						Error measures		
				Pre.	Init.	Field.	Param.	Mesh.	Post.	η_{pos}	η_{def}	η_{len}
Botijo	82332	2497	92	3.03	15.94	312.94	1379.14	4.08	0.52	0.016	0.124	0.006
Bumpy torus	33630	3426	144	3.23	36.63	100.68	445.72	2.81	0.72	0.019	0.114	0.003
Bunny	22490	2312	38	0.81	14.22	77.91	61.82	1.79	0.47	0.015	0.089	0.001
Chair	24994	5288	100	1.22	14.27	77.41	144.11	3.33	1.04	0.015	0.099	0.003
3 Holes	11776	3956	32	0.87	21.77	36.94	62.31	2.16	0.8	0.017	0.174	0.012
Elk	10388	2430	106	0.44	20.69	31.4	43.01	1.48	0.45	0.018	0.142	0.002
Fandisk	12946	2500	30	0.48	13.95	41.41	19.85	1.51	0.52	0.025	0.113	0.003
Human	13730	1792	28	0.49	14.01	39.50	15.70	1.28	0.35	0.014	0.115	0.003
Intersection	3328	817	12	0.28	13.75	12.53	1.22	0.48	0.16	0.011	0.065	0.002
Kitten	28944	2492	26	0.93	14.41	100.03	61.71	2.13	0.51	0.016	0.086	0.003
Mango	4851	2296	4	0.24	13.72	18.85	1.05	1.07	0.47	0.027	0.086	0.002
Schwarz P	6144	1799	16	0.28	13.81	19.92	3.01	1.01	0.35	0.017	0.063	0.004
Soumaya	17718	1893	6	0.59	14.09	58.92	5.43	1.45	0.38	0.017	0.069	0.004
Stent	4004	1429	4	0.22	13.69	14.66	0.93	0.72	0.28	0.010	0.051	0.004
Teddy	29806	2004	42	1.55	14.77	95.12	68.59	1.98	0.41	0.012	0.081	0.002
Train Station	3330	1097	4	0.32	14.02	13.99	0.53	0.60	0.22	0.008	0.052	0.002

Table 1. Algorithm statistics. Left to right: input number of triangles, output number of quads and singularities, timings: pre-processing (operator construction), initial solution (approximate Killing vector field), guiding field optimization, seamless parameterization, quad meshing, and post-processing ShapeUp optimization. Error measures: position error (30), deformation error (31), and edge length deviation error (32).

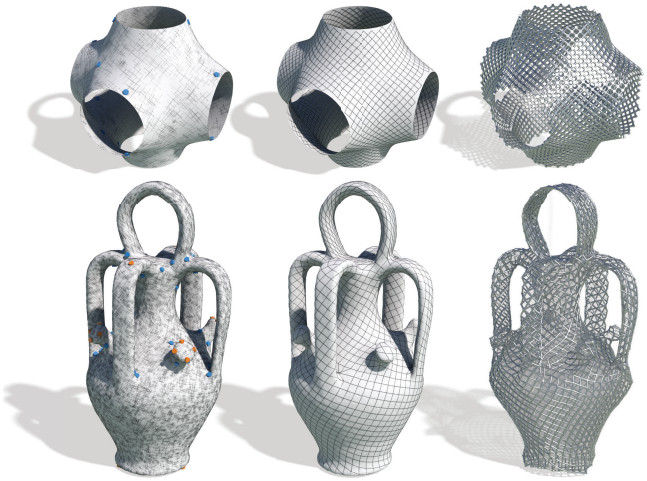


Fig. 10. Results for the Schwarz P (top) and Botijo (bottom) models. After optimizing for integrability (see Figure 9) one has guiding fields with optimal singularities (left), which is post-processed into a quad mesh (middle) that remains near the target surface after ShapeUp optimization (right).

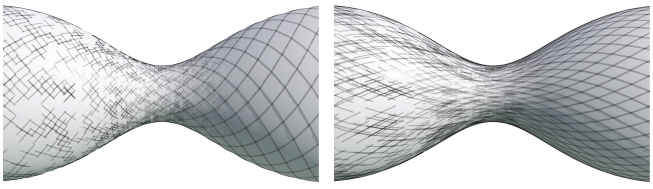


Fig. 11. The Poisson-integrated quadrangulation from unit length guiding fields that are non-integrable (left) and integrable (right). Non-integrability leads to a visible discrepancy in alignment between the guiding field and the integrated net, as shown in the middle overlapping region. Integrable unit length guiding fields integrate to Chebyshev nets.

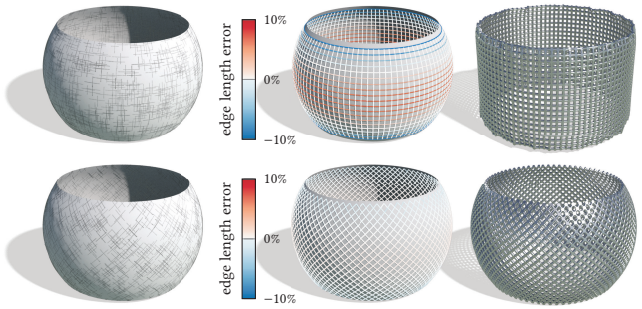


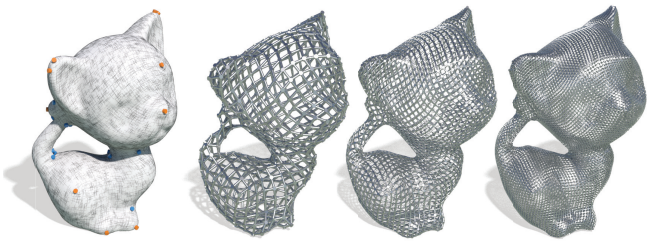
Fig. 12. Comparison between using a non-integrable field (top row) and using an integrable field (bottom row) to find Chebyshev nets. For a non-integrable guiding field, integrating into a quadrangulation does not preserve arclength along parameter lines (middle). This leads to large deviations from the target shape, when requiring unit length edges during ShapeUp optimization (right). The top-right has relative position error $\eta_{\text{pos}} = 0.0261$ and deformation error $\eta_{\text{def}} = 0.0367$, whereas the bottom-right has relative position error $\eta_{\text{pos}} = 0.0021$ and deformation error $\eta_{\text{def}} = 0.0107$.

Since our algorithm is parameterization-based, we can create Chebyshev nets of varying length scales that are effectively optimized into Chebyshev net quad meshes without considerable deformation. We find that the relative position and deformation errors correlate with visual inspection (Fig. 13).

In summary, we demonstrate that our optimization for integrability is essential to the design of Chebyshev nets.

9.4 Effect of the angle bound

A nonzero angle bound is necessary to prevent quads from collapsing to zero, while different angle bounds may reflect the particular material properties of a physically realized Chebyshev net. The space of Chebyshev nets with a specific angle-bound, however, may be too



ℓ	length scale	0.04	0.02	0.01
η_{pos}	relative position error	0.0219	0.0165	0.0145
η_{def}	deformation error	0.1180	0.0857	0.0791

Fig. 13. The results of the ShapeUp optimization on the quadrangulation obtained from the prameterization at different length scales.

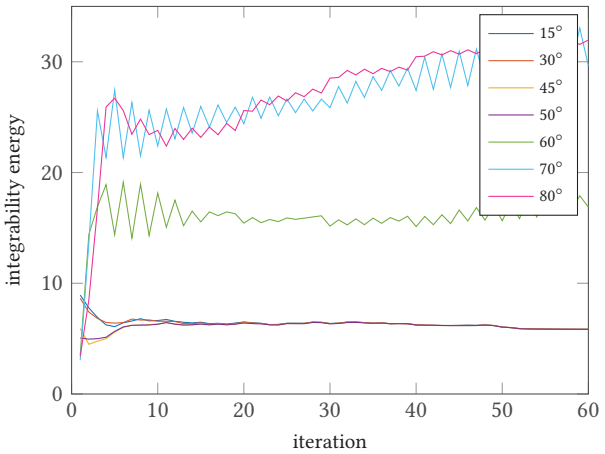


Fig. 14. Plots of the integrability energy during optimization of the truncated mango shown in Fig. 18, with varying angle bounds. Note the clear separation between the trend of the integrability energy with small angle bounds $< 50°$ versus large angle bounds $> 50°$.

restrictive for particular geometries. For example, only developable surfaces admit Chebyshev nets with all angles $\pi/2 = 90°$. In Fig. 14 we show the effect of varying the angle bound (default for all other meshes is $\pi/6$) on the integrability energy during optimization.

10 APPLICATIONS

Our algorithm can be used to automatically design Chebyshev nets for a diverse set of applications requiring a variety of complicated geometries. We present examples for architectural gridshells (Fig. 15, Fig. 16), transportation packaging (Fig. 18), medical devices (Fig. 3), and armor design (Fig. 17). Figure 19 shows the output of our algorithm on some challenging meshes from the benchmark in [Myles et al. 2014] for general quad meshing algorithms.

10.1 Physical realizability from planar pieces

To demonstrate the physical realizability of our designs, we constructed a global Chebyshev net on the Stanford Bunny (Fig. 20). To realize the net, we used a woven mesh with nearly inextensible

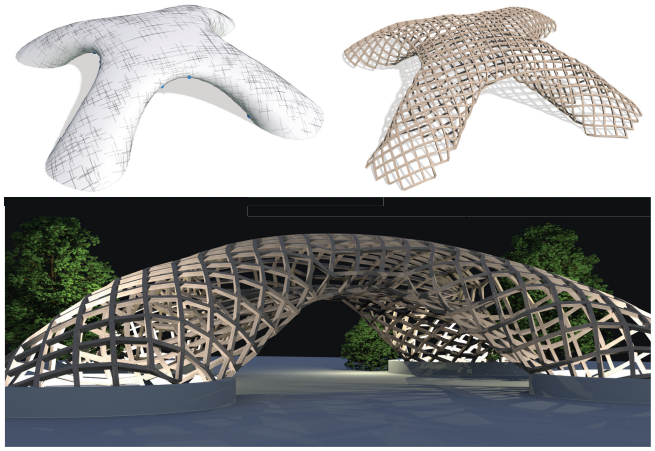


Fig. 15. An architectural gridshell designed from a Chebyshev net.

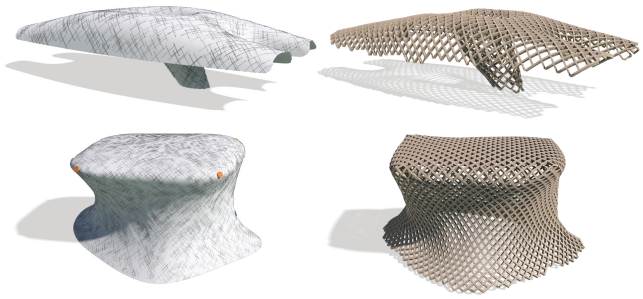


Fig. 16. Chebyshev nets automatically computed for intricate architectural models of a train station (top) and of the Museo Soumaya (bottom).

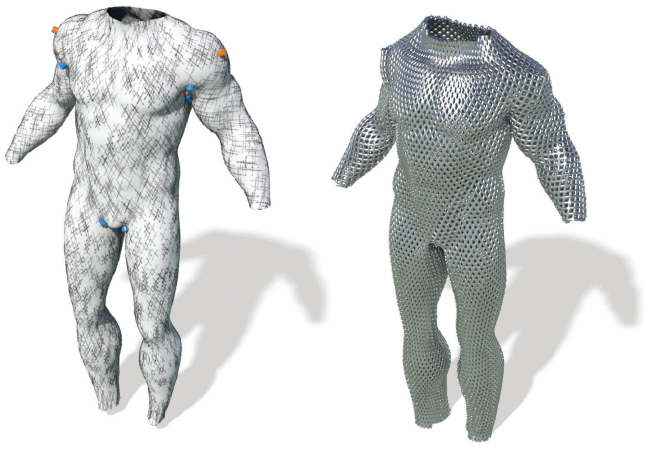


Fig. 17. A custom-made armor designed from a Chebyshev net.

yarns that can shear with respect to each other. Our algorithm computes a global Chebyshev net. Unfolding the resulting net into the plane requires that we introduce *cut lines*, which connect pairs of singularities. In practice, we use the cut lines found during the u, v

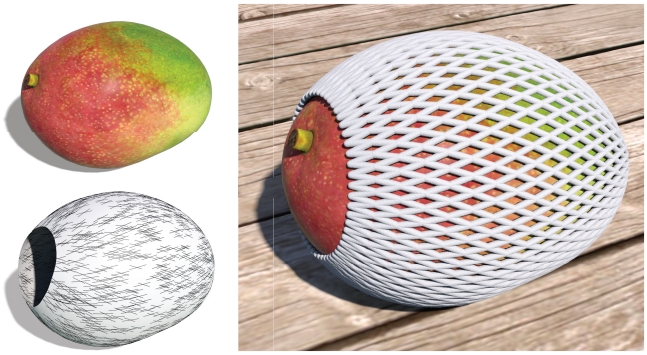


Fig. 18. Part of a mango (left) fitted with a Chebyshev net packaging (right).

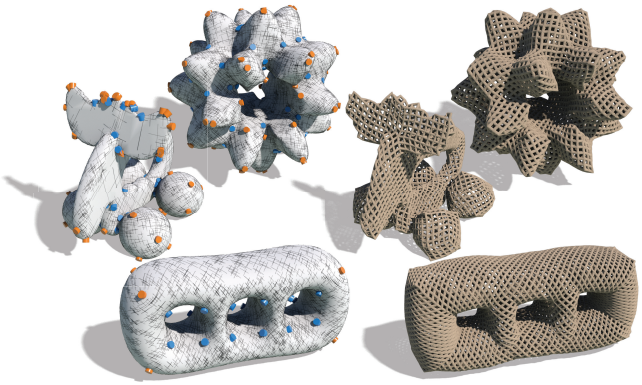


Fig. 19. Our global Chebyshev nets on quad-meshing benchmark meshes.

global parameterization step. Note that the global parameterization is seamless, and therefore the choice of specific cut lines can be made arbitrarily. There are overlapping regions in u, v space that need to be separated. We do so by manually introducing additional cut lines that separate the domain into a collection of non-overlapping regions. We obtained five pieces: one piece for the body, one for each ear, and one for each foot. Each piece was then traced onto the woven material with all singularities and cut lines marked. We left an extra layer of material around the cut lines to provide structural stability for when they were aligned and then stapled or fastened together. To realize each singularity, the material was manually rotated about the singularity by 90° and then stapled to itself on the surplus material. After assembling the five pieces into a connected multi-patch net, it was wrapped around a Stanford Bunny that was 3D-printed in two pieces and glued together. The remaining cut lines were then aligned and fastened together with medical bandage clips to form a closed surface. Note the considerable visual resemblance between the digital and the physical Chebyshev nets (Fig. 1).

11 DISCUSSION

11.1 Limitations

Our algorithm is based on a Ginzburg–Landau type of energy, and therefore encourages smooth guiding fields with low-order singularities of index $\pm 1/4$ around which most of the integrability error is concentrated. This may lead to a quad mesh with sharp corners when optimized to have unit edge length. Moreover, when $\pm 1/4$ singularities cannot suitably concentrate all integrability error, the post processing algorithm may not preserve volume. This effect can be seen both in the legs and back-support of the chairs in Fig. 22, and in the top handle of the Botijo mesh, where Fig. 9 shows before and after integrability optimization and Fig. 10 shows before and after post processing. There are Chebyshev nets on surfaces that our approach does not promote—specifically, nets that admit higher-order "rosette" singularities and non-smooth patch boundaries [Masson 2017]. We demonstrate this in Fig. 21. It would be interesting to develop methods that allow for optimizing over this larger space of Chebyshev nets with singularities.

Additionally, our discretization of the integrability energy is not fully invariant to the triangulation, and a non-uniform mesh can lead to higher integrability errors (Fig. 22). We leave further investigation of a consistent discrete formulation of the Lie bracket operator for future work. Note that despite this limitation the final Chebyshev net is mostly similar to the one obtained using a uniform mesh.

Finally, we rely on standard integer rounding of the u, v coordinates to get truly seamless (not just rotationally seamless) parameterizations that give quad meshes. As such, our integrability does not guarantee bijectivity to produce consistent quad meshes. In practice, this has not been problematic for the meshes we consider.

11.2 Future work

An essential future direction is to explore the design of fabrication-aware nets, where not only the singularities, but also the cut lines would be located in places that are convenient to manufacture and assemble. An intriguing possibility would be to design dynamic nets that flex only in desired directions. This could prove invaluable for medical applications, e.g., for the dynamic reinforcement of tissues.

ACKNOWLEDGMENTS

A.O. Sageman-Furnas and A. Chern were supported by the DFG Collaborative Research Center TRR 109 Discretization in Geometry and Dynamics. M. Ben-Chen acknowledges funding from the European Research Council (ERC starting grant No. 714776 OPREP). Additional support was provided by SideFX software.

REFERENCES

John Edward Adkins. 1956. Finite Plane Deformation of Thin Elastic Sheets Reinforced with Inextensible Cords. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 249, 961 (1956), 125–150.

Masaki Aono. 1994. *Computer-Aided Geometric Design for Forming Woven Cloth Composites.* Ph.D. Dissertation. Rensselaer Polytechnic Institute.

Masaki Aono, David E. Breen, and Michael J. Wozny. 2001. Modeling methods for the design of 3D broadcloth composite parts. *Computer-Aided Design* 33, 13 (2001), 989–1007.

Masaki Aono, Paolo Denti, David E. Breen, and Michael J. Wozny. 1996. Fitting a woven cloth model to a curved surface: dart insertion. *IEEE Computer Graphics and Applications* 16, 5 (1996), 60–70.

Omri Azencot, Mirela Ben-Chen, Frédéric Chazal, and Maks Ovsjanikov. 2013. An operator approach to tangent vector field processing. In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*. Eurographics Association, 73–82.

Omri Azencot, Maks Ovsjanikov, Frédéric Chazal, and Mirela Ben-Chen. 2015. Discrete derivatives of vector fields on surfaces–An operator approach. *ACM Transactions on Graphics (TOG)* 34, 3 (2015), 29.

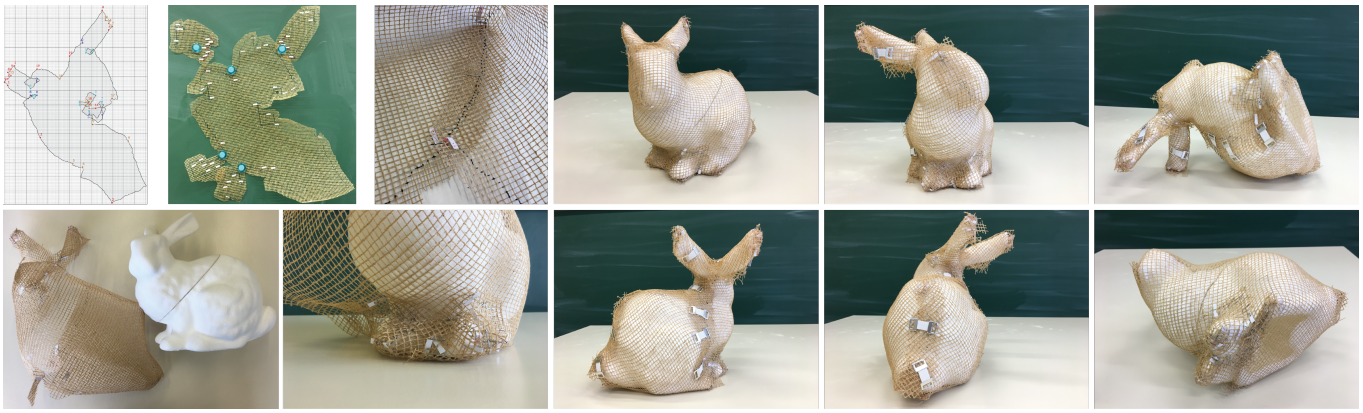


Fig. 20. Global Chebyshev net from Fig. 1, physically realized using a woven mesh that shears, but has nearly inextensible yarns. The six photos (right) show that the physical realization forms a closed surface. From a planar unfolding of our discrete Chebyshev net, we remove overlaps in the material domain by separating it into five pieces, with marked singularities and cut lines (top left). Each 1/4-singularity is then manually realized by stapling the cut lines of neighboring pieces together (top middle). The resulting multi-patch net is then draped onto the target surface (bottom left) and fastened to close the surface.

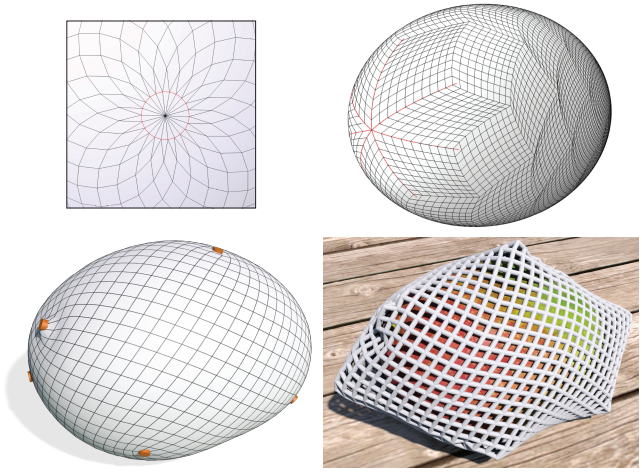


Fig. 21. A higher-order "rosette" singularity (top left) produces a piecewise-smooth Chebyshev net (top right) on an ellipsoid (both images copyright Yannick Masson [Masson 2017]). This Chebyshev net demonstrates distinct features that our algorithm penalizes. Instead, our algorithm introduces 1/4-singularities (bottom left) that lead to sharp corners in the resulting discrete Chebyshev net, giving a cube-like appearance (bottom right).

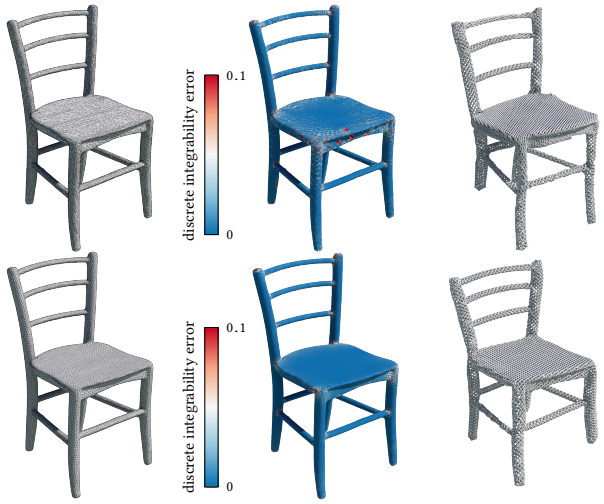


Fig. 22. Effect of the triangulation quality. The output of our algorithm on simplified (top row) and then re-meshed maintaining total number of vertices (bottom row) variants of the chair mesh from [Myles et al. 2014]. The discrete integrability energy is given by (25), and for these experiments the length scale ratio was $\ell = 0.01$.

Changyeob Baek, Andrew O Sageman-Furnas, Mohammad K Jawed, and Pedro M Reis. 2018. Form finding in elastic gridshells. *Proceedings of the National Academy of Sciences of the United States of America* 115, 1 (Jan. 2018), 75–80.

Illya Y. Bakelman. 1965. Chebyshev networks in manifolds of bounded curvature. *Trudy Matematicheskogo Instituta im VA Steklova* 76 (1965), 124–129.

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.

Fabrice Bethuel, Haïm Brezis, Frédéric Hélein, et al. 1994. *Ginzburg-Landau Vortices*. Vol. 13. Springer.

Ludwig Bieberbach. 1926. Über Tchebychefsche Netze auf Flächen negativer Krümmung, sowie auf einigen weiteren Flächenarten. *Preuss Akad Wiss, Phys Math Kl* 23 (1926), 294–321.

Bruce Blausen. 2014. Stent in Coronary Artery from Medical gallery of Blausen Medical 2014. *WikiJournal of Medicine* 1, 2 (2014). https://doi.org/10.15347/wjm/2014.010

Alexander I Bobenko and Ulrich Pinkall. 1996. Discrete surfaces with constant negative Gaussian curvature and the Hirota equation. *Journal of Differential Geometry* 43, 3 (1996), 527–611.

David Bommes, Marcel Campen, Hans-Christian Ebke, Pierre Alliez, and Leif Kobbelt. 2013a. Integer-grid Maps for Reliable Quad Meshing. *ACM Trans. Graph.* 32, 4, Article 98 (July 2013), 12 pages.

David Bommes, Bruno Lévy, Nico Pietroni, Enrico Puppo, Claudio Silva, Marco Tarini, and Denis Zorin. 2013b. Quad-Mesh Generation and Processing: A Survey. *Computer Graphics Forum* 32, 6 (2013), 51–76.

David Bommes, Henrik Zimmer, and Leif Kobbelt. 2009. Mixed-integer Quadrangulation. *ACM Trans. Graph.* 28, 3, Article 77 (July 2009), 10 pages.

Sofien Bouaziz, Mario Deuss, Yuliy Schwartzburg, Thibaut Weise, and Mark Pauly. 2012. Shape-Up: Shaping Discrete Geometry with Projections. *Computer Graphics Forum* 31, 5 (2012), 1657–1667.

Lina Bouhaya, Olivier Baverel, and Jean-François CARON. 2009. Mapping two-way continuous elastic grid on an imposed surface: Application to grid shells. In *IASS 2009: Evolution and Trends in Design, Analysis and Construction of Shell and Spatial Structures*. Valencia, Spain, 989–998.
Lina Bouhaya, Olivier Baverel, and Jean-François CARON. 2014. Optimization of gridshell bar orientation using a simplified genetic approach. *Structural and Multidisciplinary Optimization* 50, 5 (Nov. 2014), 839–848.
Alon Bright, Edward Chien, and Ofir Weber. 2017. Harmonic Global Parametrization with Rational Holonomy. *ACM Trans. Graph.* 36, 4, Article 89 (July 2017), 15 pages.
Yuri D. Burago, Sergei V. Ivanov, and Sergey G Malev. 2007. Remarks on Chebyshev Coordinates. *Journal of Mathematical Sciences* 140, 4 (2007), 497–501.
Marcel Campen, David Bommes, and Leif Kobbelt. 2015. Quantized Global Parametrization. *ACM Trans. Graph.* 34, 6, Article 192 (Oct. 2015), 12 pages.
Edward Chien, Zohar Levi, and Ofir Weber. 2016. Bounded Distortion Parametrization in the Space of Metrics. *ACM Trans. Graph.* 35, 6, Article 215 (Nov. 2016), 16 pages.
Fernando de Goes, Beibei Liu, Max Budninskiy, Yiying Tong, and Mathieu Desbrun. 2014. Discrete 2-tensor fields on triangulations. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 13–24.
Olga Diamanti, Amir Vaxman, Daniele Panozzo, and Olga Sorkine-Hornung. 2014. Designing N-PolyVector Fields with Complex Polynomials. *Computer Graphics Forum* 33, 5 (2014), 1–11.
Olga Diamanti, Amir Vaxman, Daniele Panozzo, and Olga Sorkine-Hornung. 2015. Integrable PolyVector Fields. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)* 34, 4 (2015), 38:1–38:12.
Manfredo P. do Carmo. 1992. *Riemannian Geometry*. Birkhäuser.
Cyril Douthe, Olivier Baverel, and Jean-François Caron. 2006. Form-finding of a Grid Shell in Composite Materials. *J IASS* 47 (2006), 53–62. Issue 1.
Hans-Christian Ebke, David Bommes, Marcel Campen, and Leif Kobbelt. 2013. QEx: Robust Quad Mesh Extraction. *ACM Trans. Graph.* 32, 6, Article 168 (Nov. 2013), 10 pages.
Xianzhong Fang, Hujun Bao, Yiying Tong, Mathieu Desbrun, and Jin Huang. 2018. Quadrangulation Through Morse-parameterization Hybridization. *ACM Trans. Graph.* 37, 4, Article 92 (July 2018), 15 pages.
Xiao-Ming Fu and Yang Liu. 2016. Computing Inversion-free Mappings by Simplex Assembly. *ACM Trans. Graph.* 35, 6, Article 216 (Nov. 2016), 12 pages.
Akash Garg, Andrew O Sageman-Furnas, Bailin Deng, Yonghao Yue, Eitan Grinspun, Mark Pauly, and Max Wardetzky. 2014. Wire mesh design. *ACM Trans. on Graphics* 33, 4 (July 2014), 1–12.
Étienne Ghys. 2011. Sur la coupe des vêtements: variation autour d'un thème de Tchebychev. *L'Enseignement Mathématique Revue Internationale 2e Sèrie* 57, 1–2 (2011), 165–208.
Giel, Immanuel. 2010. Multihalle — Wikipedia, The Free Encyclopedia. https://de.wikipedia.org/wiki/Multihalle
J N Hazzidakis. 1879. Über einige Eigenschaften der Flächen mit constantem Krümmungsmass. *Journal für die reine und angewandte Mathematik* 88 (1879), 68–73.
J Hennicke, K Matsushita, F Otto, K Sataka, E Schaur, T Shirayanagi, and G Gröbner. 1974. *Grid Shells (IL 10)*. Inst Lightweight Structures, Stuttgart.
Elisa Lafuente Hernández, Stefan Sechelmann, Thilo Rörig, and Christoph Gengnagel. 2013. Topology Optimisation of Regular and Irregular Elastic Gridshells by Means of a Non-linear Variational Method. In *Advances in Architectural Geometry 2012*. Springer Vienna, Vienna, 147–160.
Alec Jacobson, Daniele Panozzo, et al. 2018. libigl: A simple C++ geometry processing library. http://libigl.github.io/libigl/.
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.
John M Lee. 2013. Smooth manifolds. In *Introduction to Smooth Manifolds*. Springer, 1–31.
Yannick Masson. 2017. *Existence and construction of Chebyshev nets with conical singularities and application to gridshells*. Ph.D. Dissertation. ENPC, University Paris-Est.
Yannick Masson and Laurent Monasse. 2017. Existence of global Chebyshev nets on surfaces of absolute Gaussian curvature less than 2 pi. *Journal of Geometry* 108, 1 (01 Apr 2017), 25–32.
Ashish Myles, Nico Pietroni, and Denis Zorin. 2014. Robust Field-aligned Global Parametrization. *ACM Trans. Graph.* 33, 4, Article 135 (July 2014), 14 pages.
Ashish Myles and Denis Zorin. 2013. Controlled-distortion Constrained Global Parametrization. *ACM Trans. Graph.* 32, 4, Article 105 (July 2013), 14 pages.
Allen C. Pipkin. 1984. Equilibrium of Tchebychev nets. *Archive for Rational Mechanics and Analysis* 85, 1 (1984), 81–97.
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.
Ronald S Rivlin. 1955. Plane Strain of a Net Formed by Inextensible Cords. *Journal of Rational Mechanics and Analysis* 4 (1955), 951–974.
Ronald S. Rivlin. 1958. The deformation of a membrane formed by inextensible cords. *Archive for Rational Mechanics and Analysis* 2, 1 (1958), 447–476.
Richard E Robertson, E S Hsiue, E N Sickafus, and G S Y Yeh. 1981. Fiber rearrangements during the molding of continuous fiber composites. I. Flat cloth to a hemisphere. *Polymer composites* 2, 3 (1981), 126–131.
Sandra L. Samelson. 1991. Global Tchebychev Nets on Complete Two-Dimensional Riemannian Surfaces. *Archive for Rational Mechanics and Analysis* 114, 3 (1991), 237–254.
Sandra L Samelson and W P Dayawansa. 1995. On the Existence of Global Tchebychev Nets. *Trans. Amer. Math. Soc.* 347, 2 (1995), 651–660.
Robert Sauer. 1970. *Differenzengeometrie*. Springer Verlag, Berlin.
James J. Stoker. 1969. *Differential Geometry*. John Wiley & Sons, New York.
Pafnuty L. Tschebyscheff. 1878. Sur la coupe des vêtements, "On the cutting of garments". *Association francaise pour l'avancement des sciences* (1878), 154–155.
Betty P van West, Randolph B Pipes, and Michael Keefe. 1990. A Simulation of the Draping of Bidirectional Fabrics over Arbitrary Surfaces. *Journal of the Textile Institute* 81, 4 (1990), 448–460.
Amir Vaxman et al. 2017. Directional: directional field synthesis, design, and processing. https://doi.org/10.5281/zenodo.3338175 https://github.com/avaxman/Directional.
Amir Vaxman, Marcel Campen, Olga Diamanti, Daniele Panozzo, David Bommes, Klaus Hildebrandt, and Mirela Ben-Chen. 2016. Directional Field Synthesis, Design, and Processing. *Computer Graphics Forum* 35, 2 (2016), 545–572.
Ryan Viertel and Braxton 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.
Aurel Voss. 1882. Über ein neues Princip der Abbildung krummer Oberflächen. *Math. Ann.* 19 (1882), 1–26.
Wen-Biao Wang and Allen. C Pipkin. 1986. Inextensible Networks with Bending Stiffness. *The Quarterly Journal of Mechanics and Applied Mathematics* 39, 3 (1986), 343.
Walter Wunderlich. 1951. *Zur Differenzengeometrie der Flächen konstanter negativer Krümmung*. Springer Verlag.

A PROOF OF THEOREM 1

Since $0 < c < c_0$, the linear independence of U, V and the unit norm condition $|U|^2 = |V|^2 = 1$ follow immediately from (18). It remains to be shown that $[U, V] = 0$.

A vector field Y is Killing if and only if for all vector fields Z, W [do Carmo 1992, Chap. 3, Ex. 5d]

$$\langle \nabla_Z Y, W\rangle + \langle \nabla_W Y, Z\rangle = 0. \tag{33}$$

Therefore, for every vector field Z we have

$$\langle \nabla_Z Y, Z\rangle = 0. \tag{34}$$

In particular, $d_Y|Y|^2 = 2\langle \nabla_Y Y, Y\rangle = 0$, and by the chain rule

$$d_Y(s(|Y|)) = 0, \tag{35}$$

for all differentiable functions s of $|Y|$. Now, by substituting the definition of U, V, we compute

$$\begin{aligned}[U, V] &= [cY + s(|Y|)JY, cY - s(|Y|)JY]\\ &= -2c[Y, s(|Y|)JY]\\ &= -2cs(|Y|)[Y, JY],\end{aligned}$$

where the last equality follows from (35). Since both c and $s(|Y|)$ are nonzero, verifying $[U, V] = 0$ amounts to checking $[Y, JY] = 0$. To see that $[Y, JY] = 0$, we show that its projections onto Y and JY both vanish. On the one hand, since $\langle Y, JY\rangle = 0$ we compute

$$\begin{aligned}0 = d_Y\langle Y, JY\rangle &= \langle \nabla_Y Y, JY\rangle + \langle Y, \nabla_Y(JY)\rangle\\ &\overset{(33)}{=} -\langle \nabla_{JY} Y, Y\rangle + \langle Y, \nabla_Y(JY)\rangle = \langle \nabla_Y(JY) - \nabla_{JY} Y, Y\rangle\\ &= \langle [Y, JY], Y\rangle.\end{aligned}$$

On the other hand, we find

$$\langle [Y, JY], JY\rangle = \langle \nabla_Y(JY), JY\rangle - \langle \nabla_{JY} Y, JY\rangle = 0$$

since $\langle \nabla_Y(JY), JY\rangle = \frac{1}{2} d_Y|JY|^2 = \frac{1}{2} d_Y|Y|^2 = 0$ and $\langle \nabla_{JY} Y, JY\rangle = 0$ by (34). Therefore, $[Y, JY] = 0$, which implies that $[U, V] = 0$. □