

Chebyshev Parameterization for Woven Fabric Modeling

ANNIKA OEHRI*, ETH Zurich, Switzerland

AVIV SEGALL*, ETH Zurich, Switzerland

JING REN, ETH Zurich, Switzerland

OLGA SORKINE-HORNUNG, ETH Zurich, Switzerland

Distortion-minimizing surface parameterization is an essential step for computing 2D pieces necessary to fabricate a target 3D shape from flat material. Garment design and textile fabrication are a prominent application example. Common distortion measures quantify length, angle or area preservation in an isotropic manner, so that when applied to woven textile fabrication, they implicitly assume fabric behaves like paper, which is inextensible in all directions and does not permit shearing. However, woven fabric differs significantly from paper: it exhibits anisotropy along the yarn directions and allows for some degree of shearing. We propose a novel distortion energy based on Chebyshev nets that anisotropically penalizes shearing and stretching. Our energy formulation can be used as an optimization objective for surface parameterization and is simple to minimize via a local-global algorithm. We demonstrate its advantages in modeling nets or woven fabric behavior over the commonly used isotropic distortion energies.

CCS Concepts: • Computing methodologies → Mesh models.

Additional Key Words and Phrases: Surface parameterization, Chebyshev net, Chebyshev parameterization, woven fabric modeling

ACM Reference Format:

Annika Oehri, Aviv Segall, Jing Ren, and Olga Sorkine-Hornung. 2024. Chebyshev Parameterization for Woven Fabric Modeling. *ACM Trans. Graph.* 43, 6, Article 205 (December 2024), 15 pages. <https://doi.org/10.1145/3687928>

1 INTRODUCTION

Distortion-minimizing surface parameterization is a core problem in computer graphics. Computing a flattening, i.e., a mapping between a disk-topology surface patch and the uv -plane, is essential for texture mapping, and it is also commonly used for remeshing and shape approximation by parametric patches and other means that require a planar reference domain. The quality of the flattening crucially affects the downstream applications that rely on the parameterization, so designing suitable distortion measures and effective optimization methods is key [Hormann et al. 2007].

An important application of piecewise surface flattening is the physical fabrication of a target shape from flat pieces of sheet material, such as paper, fabric, plush, rubber or sheet metal. In this scenario, the flattening needs to mimic the behavior of the sheet

*joint first authors with equal contribution

Authors' addresses: Annika Oehri, ETH Zurich, Switzerland, oehria@ethz.ch; Aviv Segall, ETH Zurich, Switzerland, aviv.segall@inf.ethz.ch; Jing Ren, ETH Zurich, Switzerland, jing.ren@inf.ethz.ch; Olga Sorkine-Hornung, ETH Zurich, Switzerland, sorkine@inf.ethz.ch.

Permission to make digital or hard copies of part or all 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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2024 Copyright held by the owner/author(s).

0730-0301/2024/12-ART205

<https://doi.org/10.1145/3687928>



Fig. 1. Food and toy meshes with Chebyshev net protection computed by our method. The meshes are cut into disk topology (seams marked in blue).

material in order for the fabricated object to approximate the target geometry well. Hence the distortion measure optimized by the flattening should model the properties of the sheet material. Distortion energies are also crucial for segmenting the target shape into pieces that are likely to be accurately realizable by forming the intended material, either in a preprocess (e.g., [Decaudin et al. 2006; Julius et al. 2005]) or interleaved with the flattening itself (e.g., [Li et al. 2018; Poranne et al. 2017]).

Widely used distortion metrics measure deviation from isometry or conformality. Optimization of isometry strives to preserve both lengths and angles on the surface, uniformly in all directions. In the context of sheet materials, this is useful for papercraft or sheet metal bending—materials that prohibit any stretching and require nearly perfect isometry—or when using elastic materials like rubber or elastane-based knitted textiles, which can stretch equally in any direction. Conformal methods strive to preserve angles and typically allow for uniform scaling, which corresponds to auxetic behavior and is instrumental in metamaterial design [Konaković et al. 2016].

While different distortion energies proposed over the years offer various combinations of penalties on length, angle and area distortion, they do so *isotropically* for the most part. Woven fabric is a ubiquitous sheet material that is *anisotropic*: the yarns forming a grid are largely inextensible, but the textile can stretch by a limited amount diagonally to the yarn directions (see Fig. 2). Many works employ distortion energy based mesh segmentation and parameterization to design sewing patterns for textile fabrication [Decaudin et al. 2006; Julius et al. 2005], but they essentially cast the problem as developability optimization, treating fabric similarly to paper [Rose et al. 2007] and not accounting for its anisotropic stretch capacity.

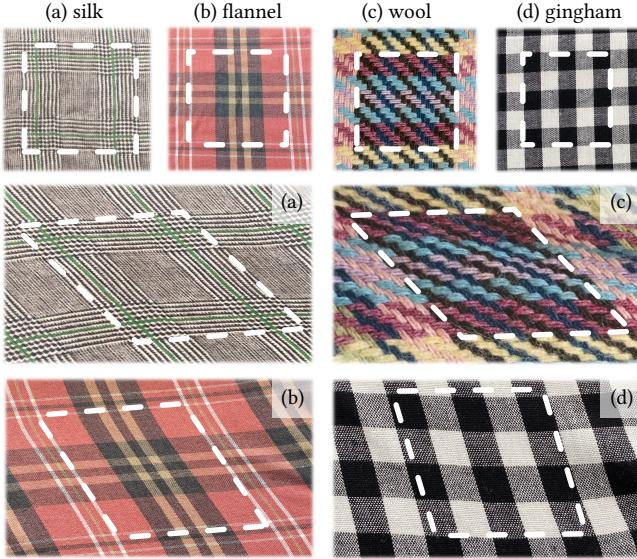


Fig. 2. Four fabrics with different materials in rest state (top): (a) 100% silk, (b) flannel (twill weave, 100% cotton), (c) 80% wool with 20% polyester, and (d) gingham (plain weave, 100% cotton). All these yarns are almost non-stretchable, thus when the fabric is deformed by external forces, shearing occurs, as illustrated in the middle and bottom rows. Note that the camera view is parallel to the fabric plane.

Modeling such anisotropy and formulating a suitable parameterization distortion energy is a challenging inverse problem, because the yarn directions are unknown *a priori*. As a result, digital garment design and other textile fabrication applications are largely based on flat sewing patterns and developable formulations, and the fabric material properties are typically considered during the simulation phase, rather than the modeling phase, necessitating trial-and-error loops of sewing pattern editing and 3D draping simulation [CLO Virtual Fashion 2022].

Chebyshev [1878] initially introduced a “cable-net” structure to parameterize a piece of fabric that assumes no stretch along the *uv*-lines (the warp and weft directions) while allowing them to shear, i.e., the angles between the *uv* lines do not necessarily have to be 90°. This unique net structure gained interest in the geometry community, particularly as a tool for modeling architectural structures or wired nets [Garg et al. 2014; Liu et al. 2020; Masson and Monasse 2017; Montagne et al. 2020; Sageman-Furnas et al. 2019]. However, despite its original intention, adopting Chebyshev nets for surface parameterization in textile fabrication scenarios is not common due to the mentioned challenges. Pietroni et al. [2022] consider the anisotropic behavior of fabric when computing 2D sewing patterns for input 3D garments, in an attempt to model fabric as a Chebyshev net, but their method lacks a proper continuous energy discretization and suffers from stability issues, as we discuss in Sec. 5.

Our main contribution is the characterization of a Chebyshev parameterization through its Jacobian. We analyze the Jacobian via its singular value decomposition and reveal its particular structure, enabling us to formulate a Chebyshev distortion energy that measures the deviation of a given parameterization from a Chebyshev net. We develop an efficient algorithm, which is partially based on

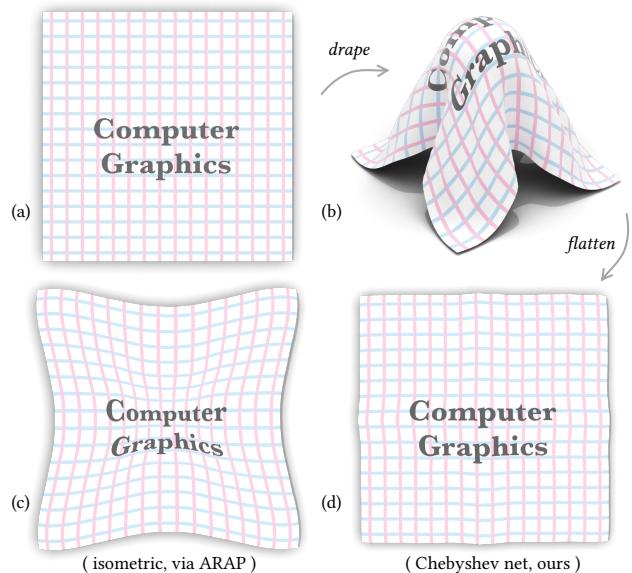


Fig. 3. A piece of fabric (a) is draped onto a sphere using Blender [2024], resulting in the shape (b). Next, we flatten the 3D draped cloth by minimizing either the isometric energy [Liu et al. 2008] (c) or our Chebyshev net energy (d). We can see that the Chebyshev net more effectively models woven fabric and can more accurately recover its original state during flattening.

the Procrustes problem, to minimize our proposed Chebyshev distortion measure and demonstrate that it produces flattening mappings that accurately reflect the qualitative behavior of woven textiles, wire meshes and other similar sheet materials (Fig. 1). Our results indicate that the proposed Chebyshev energy is better suited than the standard conformal or isometric energies in such scenarios. We also show that the same Chebyshev Jacobian characterization can be used as a component in a shape deformation energy for freeform interactive editing of shapes that are intended to be realized with fabric, in a similar spirit to how developability measures are used for freeform design of folded paper shapes [Rabinovich et al. 2018].

2 RELATED WORK

We review prior research related to mesh parameterization and Chebyshev nets, as well as modeling woven fabric.

Chebyshev nets. Chebyshev [1878] introduced a net structure to represent a piece of fabric based on the inextensibility property of yarns placed along the warp and weft directions that comprise woven fabric. This model, known today as *Chebyshev net*, is widely explored in applications such as wired mesh modeling for fabrication and architectural design [Garg et al. 2014; Liu et al. 2020; Montagne et al. 2020; Sageman-Furnas et al. 2019], as well as woven fabric fitting in the textile industry [Aono et al. 2001, 1994, 1996; Ramgulam 2001; Trochu et al. 1996; van West et al. 1990; Wang et al. 2005]. The woven cloth fitting problem [Mack and Taylor 1956] explores how a piece of fabric can tightly wrap a given 3D shape, and it involves computing the placement of warp and weft yarns on the surface, or their crossing points, i.e., the nodes of the woven net, assuming each point on the cloth is in contact with the surface.

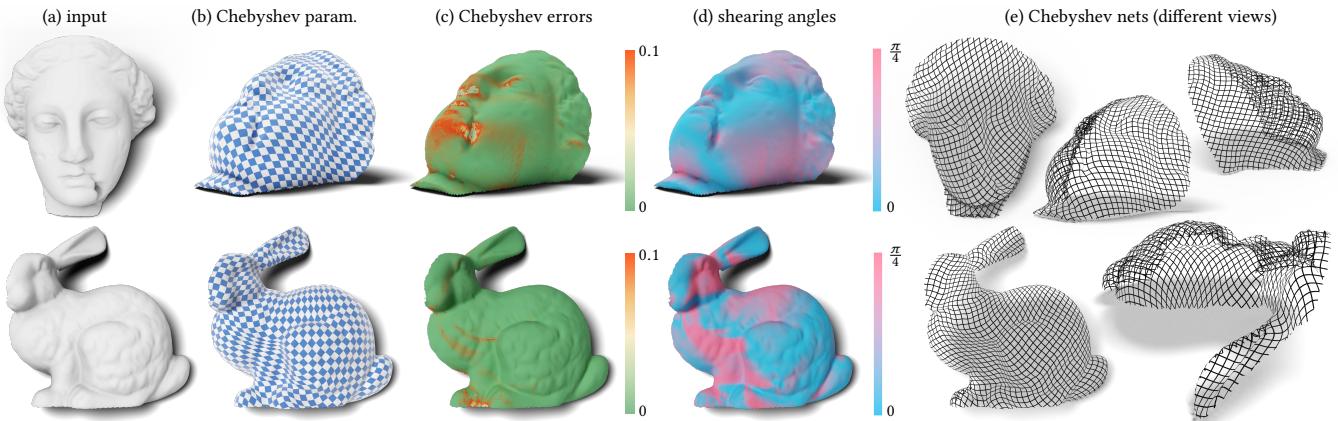


Fig. 4. We can easily extract a Chebyshev net (e) from the computed Chebyshev parametrization (b) to approximate the input 3D surface (a). Here we show two examples similar to those shown in [Garg et al. 2014, Fig. 10]. We also visualize the per-face Chebyshev error (c) and the shearing angle (d).

Mack and Taylor [1956] established the fundamental assumptions for the woven fabric fitting problem, which can be viewed as ensuring that the yarns form a Chebyshev net. They explored analytical solutions for fitting fabric to surfaces of revolution. Robertson et al. [1981, 1984] proposed the first computational method to find the 3D positions of the yarn crossing points when wrapping cloth on spherical and cone surfaces. Van West et al. [1990] were the first to consider fitting woven fabric to arbitrary surfaces while accommodating constrained yarn paths, where two perpendicular yarn curves aligned with the warp and weft directions are specified. Aono et al. [1994] simplified the process of specifying initial conditions for constrained yarns by taking just one arbitrary curve as user input. This line of research aims to determine the 3D position of each yarn crossing point after draping by finding the intersection between the input surface and two spheres, each representing a warp and a weft segment, which is computationally expensive and sensitive to the initial guess. Wang et al. [2005] model woven fabric using a mass-spring system, where the diagonal springs have a lower spring constant compared to the warp and weft springs. As a result, the deformation of the diagonal springs drives the overall deformation, mimicking shear. The woven fabric fitting problem is thereby reduced to the minimization of the strain energy defined on the mass-spring system, resulting in a more efficient and robust algorithm. However, the spring constants are chosen empirically, lacking a more precise mathematical formulation for the shearing deformation of woven fabric.

Another related topic is Chebyshev net deformation. Li et al. [2022] allows users to deform a given Chebyshev net (represented by a quad mesh) by dragging along handles while preserving edge lengths. In this work, we propose a unified framework for modeling and deforming Chebyshev nets. Other topics related to Chebyshev nets, including reconstruction from a single view [Koenderink and van Doorn 1998] and their existence for arbitrary surfaces [Masson and Monasse 2017], are outside the scope of our research.

Surface parameterization. We refer the interested readers to the thorough surveys [Floater and Hormann 2005; Hormann et al. 2007; Sheffer et al. 2007] for more detailed discussions. Here we mainly focus on surface parameterization methods used in applications

involving cloth modeling. For most shape approximation tasks using fabric, such as designing plush toys [Mori and Igarashi 2007], covers [Igarashi et al. 2009], formwork [Zhang et al. 2019], or garments [Decaudin et al. 2006; McCartney et al. 1999; Pietroni et al. 2022], the target 3D surface is usually (piecewise) flattened to create a 2D sewing pattern by using a distortion-minimizing parameterization method. The necessary decomposition of the 3D shape into patches with disk topology that can be flattened with low distortion is guided either by the same distortion measure as the subsequent parameterization [Sorkine et al. 2002] or by a developability measure [Julius et al. 2005]. These distortion measures typically *isotropically* penalize deviation from isometry (e.g., [Liu et al. 2008; Sander et al. 2001; Sorkine et al. 2002]), effectively modeling cloth as rubber material that equally resists stretching in every direction; zero distortion energy is attainable only on developable surfaces. Choosing such distortion energy disregards the anisotropic nature of woven fabric, which is nearly inextensible along the yarn directions and less resistant to stretch in other directions. Conformal or angle-preserving mappings [Lévy et al. 2002; Sawhney and Crane 2017; Sheffer et al. 2005], which preserve angles at the cost of area distortion, are suitable for modeling deployable structures with isotropic expansion ratios, such as auxetics [Konaković-Luković et al. 2018], but they are likewise less suitable for addressing the anisotropic behavior of woven fabric.

In this work, we aim to mathematically formulate and automatically compute *Chebyshev parameterizations* to express the fabric anisotropy. McCartney et al. [2005, 2000] adopt *affine* transformations to describe the anisotropic distortion of woven fabric during flattening. Affine transformations describe the parallel alignment of the weft/warp yarns, with shearing considered, but affine shears do not preserve lengths. McCartney et al.'s methods involve costly non-linear energy minimization and permit stretch along the yarn directions. In contrast, we introduce a different type of distortion energy that is designed to preserve lengths along yarn directions and can be optimized efficiently. Wang [2007] attempts to flatten woven fabric while preserving the lengths of specified feature curves along the yarn directions, but they need to be known a priori. Pietroni et al. [2022] compute a Chebyshev-like parameterization by discretely

sampling three yarn points per mesh triangle and representing yarn directions by finite differences based on the sampled points. While this approach can yield a good approximation, it does not provide a proper discretization formulation to compute Chebyshev nets reliably, leading to numerical instabilities and diverging results.

Global parameterization techniques have been explored to compute Chebyshev nets [Liu et al. 2020; Sageman-Furnas et al. 2019] by optimizing a unit polyvector field over the target surface that can be integrated into a Chebyshev net. However, these symmetric field based methods are costly and introduce singularities in the net, which do not occur in standard woven fabric. In this work we provide an efficient method to compute regular Chebyshev nets on disk topology surfaces.

Cloth simulation. Fabric modeling and simulation are closely related; modeling a target shape using a Chebyshev net can be viewed as draping a piece of woven fabric over the target surface [Breen 2000]. In simulation, cloth is often modeled as an elastic material [Baraff and Witkin 1998; Choi and Ko 2002; Liu et al. 2013; Narain et al. 2012] that allows isotropic stretch. Some simulators account for the anisotropic behavior of woven fabric, e.g. particle based models [Breen et al. 1994], mass-spring systems with inextensibility constraints [Goldenthal et al. 2007], piecewise linear elastic models [Wang et al. 2011], hyper-elastic constitutive models [Peng et al. 2013] and yarn-level approaches [Cirio et al. 2014; Kaldor et al. 2010], which exhibit exceptional accuracy at a great computational cost. Cloth simulation methods require a flat rest state and known warp and weft directions. In other words, they solve the problem of deforming a flat piece of cloth onto a 3D surface, whereas we are after the *inverse* problem of computing a mapping from the 3D surface onto the flat fabric domain, without a priori knowledge of the fabric grain directions.

3 CHEBYSHEV PARAMETERIZATION

3.1 Chebyshev net & its Jacobian

The Chebyshev net is a parametric surface representation, where there is no stretch along the parameter lines [Chebyshev 1878]. Specifically, let \mathcal{M} denote an oriented 2-manifold in 3D with disk topology that is parameterized by $f : \mathbb{R}^2 \rightarrow \mathcal{M}$, i.e., each point (x, y, z) on \mathcal{M} is expressed as $f(u, v)$. We call f a *Chebyshev parameterization* if

$$\left\| \frac{\partial f}{\partial u} \right\| = 1, \quad \left\| \frac{\partial f}{\partial v} \right\| = 1. \quad (1)$$

Let n denote the outward surface normal according to \mathcal{M} 's orientation. Select an arbitrary orthonormal basis of the tangent plane, $\{t_1, t_2\}$, such that (t_1, t_2, n) form a right-hand system. We express the tangent vectors in this basis: $f_u = [\partial f / \partial u]_{(t_1, t_2)}$, $f_v = [\partial f / \partial v]_{(t_1, t_2)}$. The Jacobian of f is a linear map from the parametric uv -domain to the tangent plane at $f(u, v)$ that maps the u, v axes to the tangent vectors f_u, f_v . The Jacobian of a Chebyshev net can be expressed as a 2×2 matrix $J = (f_u \ f_v)$ that has a simple form based on the above inextensibility constraints Eq. (1): J consists of two column vectors of unit length. We denote the space of all such 2×2 matrices as \bar{C} :

$$\bar{C} = \left\{ \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} \mid c_{11}^2 + c_{21}^2 = 1, \ c_{12}^2 + c_{22}^2 = 1 \right\}. \quad (2)$$

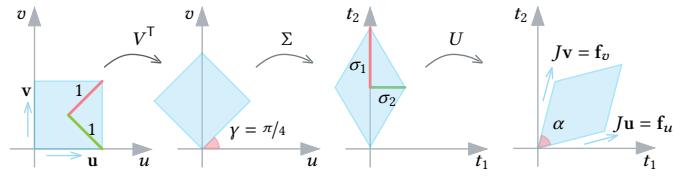


Fig. 5. Illustration of the singular value decomposition $J = U\Sigma V^\top$, where J is the Jacobian of a Chebyshev net on \mathcal{M} . We denote the *yarn angle* on \mathcal{M} , i.e., the angle between the two yarn directions J_u and J_v , as α .

In addition to the inextensibility property, we assume that f agrees with the given orientation on \mathcal{M} . Therefore, we focus on a subset of Chebyshev nets that preserve orientation, $C = \{J \in \bar{C} \mid \det(J) > 0\}$.

3.2 Singular value decomposition of J

Denote the SVD of the Jacobian of an orientation-preserving Chebyshev net as $J = U\Sigma V^\top$. We analyze its structure and make the following observations:

- The V matrix can always be chosen to be a rotation matrix representing a clockwise rotation by $\pi/4$, which we denote as $R_{\pi/4}$.
- Since in our setting $\det J > 0$ and the singular values are always non-negative, the U matrix is a rotation with $\det U > 0$.
- The two singular values of Σ always lie on a circle with radius $\sqrt{2}$, i.e., $\sigma_1^2 + \sigma_2^2 = 2$.
- The determinant of the Jacobian is equal to the sine of the yarn angle α (defined in Fig. 5), i.e., $\det(J) = \sin \alpha$.

Therefore, we can rewrite the search space for Jacobians of a Chebyshev parameterization as:

$$C = \left\{ U \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} R_{\pi/4}^\top \mid \sigma_1^2 + \sigma_2^2 = 2, \ \sigma_1, \sigma_2 > 0, \ U \in SO(2) \right\}. \quad (3)$$

Refer to Appendix A for detailed proofs and further mathematical observations. Here we provide an intuitive explanation, illustrated in Fig. 5. The Jacobian of a Chebyshev parameterization is a linear transformation that preserves lengths along the u and v directions while allowing the angles to shear, which can be regarded as transforming a square into a diamond shape (an equilateral parallelogram) by moving the vertices along the square's diagonals. The singular value decomposition of J can be interpreted as follows (see Fig. 5): First, rotate the square counterclockwise by $\pi/4$, such that its diagonals are aligned with the coordinate axes. Then, the rotated square is stretched along the diagonals, with the scaling factors being the singular values. The singular values should satisfy the constraint $\sigma_1^2 + \sigma_2^2 = 2$ to make sure that the edge length of the diamond shape is the same as in the initial square. Lastly, another rotation U is applied to transform the diamond such that its edge vectors correctly align with the yarn directions, i.e., with the derivatives of the parameterization. We can easily see that the determinant of the Jacobian is the area of the diamond shape, which corresponds to the sine of the yarn angle α .

4 METHOD

For a given oriented triangle mesh $\mathcal{M} = (\mathcal{V}, \mathcal{T})$ of disk topology with vertices $\mathcal{V} \subset \mathbb{R}^3$ and triangle faces \mathcal{T} , a common strategy to obtain its parameterization is to flatten the input 3D surface onto the 2D domain by finding a function $g : \mathcal{M} \rightarrow \mathbb{R}^2$; its inverse g^{-1}

is the parameterization function. We call g a *Chebyshev flattening* if g^{-1} is a Chebyshev parameterization. We consider piecewise linear functions g , described by their values on the mesh vertices \mathcal{V} , with the values in the triangle interiors linearly interpolated using barycentric coordinates. Picking an arbitrary local orthonormal basis on triangle $t \in \mathcal{T}$, we denote the corresponding 2×2 Jacobian of g on t as B_t . Since g^{-1} is a Chebyshev parameterization, as discussed earlier, its Jacobian $J_t \in \mathbb{C}$ can be written as $J_t = U\Sigma R_{\pi/4}^T$, where U is a rotation matrix, and the sum of squared diagonal entries in Σ is 2. We then know that for the Jacobian B_t of the flattening g :

$$B_t = J_t^{-1} = \left(U\Sigma R_{\pi/4}^T \right)^{-1} = R_{\pi/4}\Sigma^{-1}U^T. \quad (4)$$

Therefore, if we apply singular value decomposition to B_t , the matrix of the *left* singular vectors should be a rotation by $\pi/4$, and the two singular values of B_t , denoted by λ_1 and λ_2 , should satisfy $1/\lambda_1^2 + 1/\lambda_2^2 = 2$, while the matrix of the *right* singular vectors is some rotation matrix. We can therefore define the *search space* for the Jacobians of a Chebyshev flattening as

$$\mathcal{B} = \left\{ R_{\pi/4} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} V^T \mid \frac{1}{\lambda_1^2} + \frac{1}{\lambda_2^2} = 2, \lambda_1, \lambda_2 > 0, V \in SO(2) \right\}. \quad (5)$$

Now we define the following energy that measures how close g is to a Chebyshev flattening:

$$E(g) = \sum_{t \in \mathcal{T}} A_t \mathcal{D}(B_t, \mathcal{B})^2, \quad (6)$$

where A_t is the area of face $t \in \mathcal{T}$, and $\mathcal{D}(B_t, \mathcal{B})$ is the distance between the Jacobian B_t and the set \mathcal{B} (defined in Eq. (5)):

$$\mathcal{D}(B_t, \mathcal{B}) = \min_{Y \in \mathcal{B}} \|B_t - Y\|_F, \quad (7)$$

where $\|\cdot\|_F$ is the Frobenius matrix norm. Note that $\mathcal{D}(B_t, \mathcal{B})$ is invariant to the choice of the local orthonormal frame on t . We call

$$Y_t = \arg\min_{Y \in \mathcal{B}} \|B_t - Y\|_F \quad (8)$$

the *closest Chebyshev-Jacobian* to B_t . To find a Chebyshev flattening g of a given mesh, we need to minimize the non-linear, non-convex energy in Eq. (6), where the variables are the (u, v) -coordinate assignments for each mesh vertex. To do so, we adopt a local-global strategy similar to ARAP [Liu et al. 2008; Sorkine and Alexa 2007]:

- (1) **Initialization:** g is initialized by a least-squares conformal flattening [Lévy et al. 2002], and we compute its Jacobians B_t for each face $t \in \mathcal{T}$.
- (2) **Local step:** compute Eq. (8) with fixed B_t . For each face $t \in \mathcal{T}$, we compute Y_t , the closest Chebyshev-Jacobian to B_t .
- (3) **Global step:** solve Eq. (6) while fixing the known closest Chebyshev-Jacobian, i.e., solve $\min_Y \sum_{t \in \mathcal{T}} A_t \|B_t - Y_t\|_F^2$.
- (4) Go to step (2) until convergence.

Computing the closest Chebyshev-Jacobian in the local step is challenging, since the search space defined in Eq. (5) is quite complicated. We decompose the local step into two minimization tasks: first, solve for the best-fit singular values for a given matrix of right singular vectors V , and second, solve for the best-fit rotation matrix $V \in SO(2)$ for the given singular values.

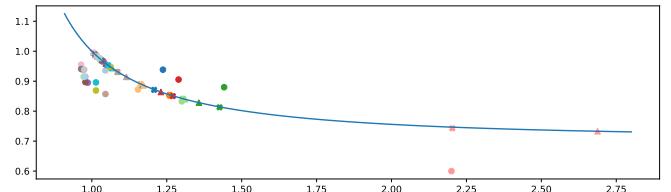


Fig. 6. We draw the curve $\ell : 1/\lambda_1^2 + 1/\lambda_2^2 = 2$ in blue and show the estimated singular values from different starting points (n_{11}, n_{22}) in different colors. Points in *dot* shape are the starting points (n_{11}, n_{22}) . Points in *triangle* shape are the initial guess $(1/x, 1/y)$, and the points in *cross* shape are the final estimation, which is close enough to the expected curve.

4.1 Solving the local step: finding the best Σ_t

With fixed Jacobian B_t and right singular vectors V_t , we aim to solve for the closest singular values, namely λ_1, λ_2 that satisfy $1/\lambda_1^2 + 1/\lambda_2^2 = 2$:

$$\Sigma_t = \underset{\substack{\frac{1}{\lambda_1^2} + \frac{1}{\lambda_2^2} = 2}}{\operatorname{argmin}} \|B_t - R_{\pi/4}\Sigma V_t^T\|_F^2 = \underset{\substack{\frac{1}{\lambda_1^2} + \frac{1}{\lambda_2^2} = 2}}{\operatorname{argmin}} \|R_{\pi/4}^T B_t V_t - \Sigma\|_F^2. \quad (9)$$

Denote $N = R_{\pi/4}^T B_t V_t$ with diagonal values n_{11} and n_{22} . The optimal singular values can be interpreted as the 2D point (λ_1, λ_2) lying on the 2D curve ℓ , defined as $1/\lambda_1^2 + 1/\lambda_2^2 = 2$, that is closest to the given position (n_{11}, n_{22}) . In Fig. 6, we draw the expected curve ℓ in blue, and illustrate the diagonal values as 2D points (n_{11}, n_{22}) in *dot* shape. Solving this algebraically yields a polynomial of high degree that does not have a simple closed-form solution. We propose the following algorithm to find good approximations. We first solve a related but simpler problem: finding the 2D point (x, y) lying on the 2D circle $x^2 + y^2 = 2$ that is closest to the given position $(1/n_{11}, 1/n_{22})$, which is the intersection point between the 2D circle and a line passing through $(0, 0)$ and $(1/n_{11}, 1/n_{22})$. Once we get the intersection point (x, y) , we obtain an initial guess for the singular values, i.e., $\lambda_1 = 1/x$, $\lambda_2 = 1/y$. We draw the points $(1/x, 1/y)$ in *triangle* shape in Fig. 6. Though this estimation lies on the expected curve $\ell : 1/\lambda_1^2 + 1/\lambda_2^2 = 2$, it is not necessarily the closest projection w.r.t. (n_{11}, n_{22}) . To improve it, we derive the first-order Taylor approximation of the curve ℓ at the estimated position $(1/x, 1/y)$, leading to a 2D line ℓ' . We then project $(1/n_{11}, 1/n_{22})$ onto ℓ' , which gives us the final estimated singular values, i.e., the points in *cross* shape in Fig. 6.

4.2 Solving the local step: finding the best V_t

Having computed the singular values Σ_t , we can now update the matrix of right singular vectors, i.e., the rotation V accordingly:

$$V_t = \underset{V \in SO(2)}{\operatorname{argmin}} \|B_t - R_{\pi/4}\Sigma_t V^T\|_F^2. \quad (10)$$

This problem is equivalent to the orthogonal Procrustes problem, which has a simple solution via singular value decomposition: denote $S = \Sigma_t R_{\pi/4}^T B_t$ and apply SVD to $S = HDK^T$, then the optimal V_t is given by $V_t = KH^T$. In case $\det(KH^T) < 0$, we multiply the column of K corresponding to the smallest singular value by -1 .

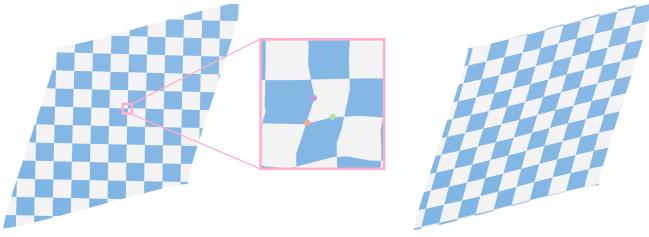


Fig. 7. Left: A sheared square mesh after parameterizing it with constraints via a least-square conformal map [Lévy et al. 2002]. Middle: Closeup of the used constraints, three vertices on a single triangle are used to define the two yarn directions, with one fixed point (orange), one point being constrained to be on the same yarn as it is in warp (green) and weft (purple) direction, respectively. Right: Chebyshev parameterization with the same constraints.

4.3 Solving the global step

In the discrete setting, the piecewise linear flattening function g is represented by a vector of $2n$ unknowns, (\mathbf{u}, \mathbf{v}) , where $n = |\mathcal{V}|$, i.e., by the per-vertex uv -coordinates. The computed local step provides us with good approximations of the closest Chebyshev-Jacobians, i.e., $Y_t = R_{\pi/4} \Sigma_t V_t^\top$ for each mesh triangle $t \in \mathcal{T}$, and we want to solve for the (\mathbf{u}, \mathbf{v}) by minimizing:

$$\min_{(\mathbf{u}, \mathbf{v})} \sum_{t \in \mathcal{T}} A_t \|B_t(\mathbf{u}, \mathbf{v}) - Y_t\|_F^2, \quad (11)$$

where $B_t(\mathbf{u}, \mathbf{v})$ is the (constant) Jacobian of g on triangle t . This is exactly the setting described in [Liu et al. 2008, Sec. 4.4.2], and the uv -coordinates are obtained by solving a sparse linear system coming from the derivative of the energy in Eq. (6). The system matrix is essentially the cotangent Laplacian of the input mesh, and it is fixed in our problem, such that its sparse factorization can be pre-computed and reused in each iteration, since only the right-hand-side vector in the linear system changes.

We can also easily add positional constraints to this linear system. This can help to incorporate known yarn directions or constrain parametric boundaries to be straight. For example, in garment sewing patterns, certain patch boundaries and cuts often need to be straight and aligned with the grain direction. To integrate this knowledge into the global step, one can e.g. constrain some (or all) vertices along a given curve on the mesh to have the same u coordinate in the parameter space, while the v coordinate is still free to be optimized over. An example where this approach is used is shown in Fig. 7, where a sheared square is parameterized, and by adding constraints to a single triangle, the shearing can be recovered when using our method, while the same constraints have very little effect on the result of the conformal map.

4.4 Chebyshev error

Having optimized the Chebyshev flattening function $g : \mathcal{M} \rightarrow \mathbb{R}^2$ with its Jacobians $B_t, t \in \mathcal{T}$, we quantify the Chebyshev error of g as follows:

$$e_{\text{cheby}}(g) = \sum_{t \in \mathcal{T}} A_t \mathcal{D}(B_t^{-1}, C)^2, \quad (12)$$

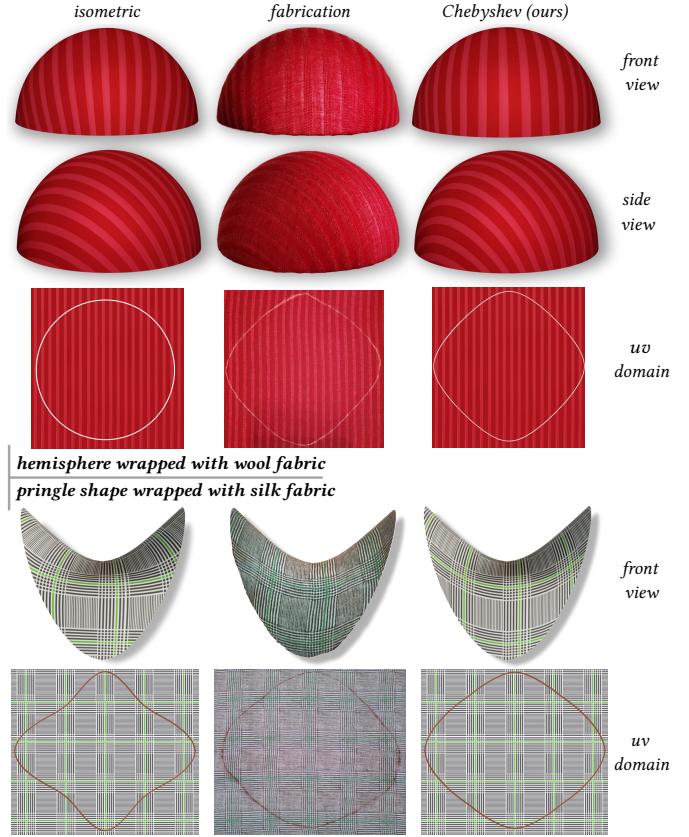


Fig. 8. To find the ground-truth Chebyshev parameterization for a hemisphere (top) and pringle shape (bottom), we tightly wrap the 3D-printed shapes with a piece of wool and silk, respectively (middle column) and mark the boundary using chalk. The same (digital) shape is flattened via ARAP [Liu et al. 2008] (left) and our Chebyshev formulation (right). Our flattening result shows a much closer correspondence to the fabricated result than the isometric approach, both visible on the 3D model and the marked outline in the uv domain.

since this formulation provides a more direct evaluation on the deviation of g^{-1} from a Chebyshev net and does not require computing the SVD decomposition during evaluation: the closest element to B_t^{-1} in C is given by simply normalizing its columns according to Eq. (2), and their distance measures the stretch along the warp and weft directions (denoting the two columns of B_t^{-1} as J_1, J_2):

$$\mathcal{D}(B_t^{-1}, C)^2 = \left\| J_1 - \frac{J_1}{\|J_1\|} \right\|^2 + \left\| J_2 - \frac{J_2}{\|J_2\|} \right\|^2 = (\|J_1\| - 1)^2 + (\|J_2\| - 1)^2.$$

5 RESULTS

In this section, we demonstrate that our Chebyshev parameterization (1) offers a more accurate and faithful model for 3D shapes realized with woven fabric (Sec. 5.1), (2) supports practical applications such as Chebyshev net construction (Sec. 5.2) and Chebyshev deformation (Sec. 5.3) and (3) is parameter-free, straightforward to implement and highly efficient (Sec. 5.4).

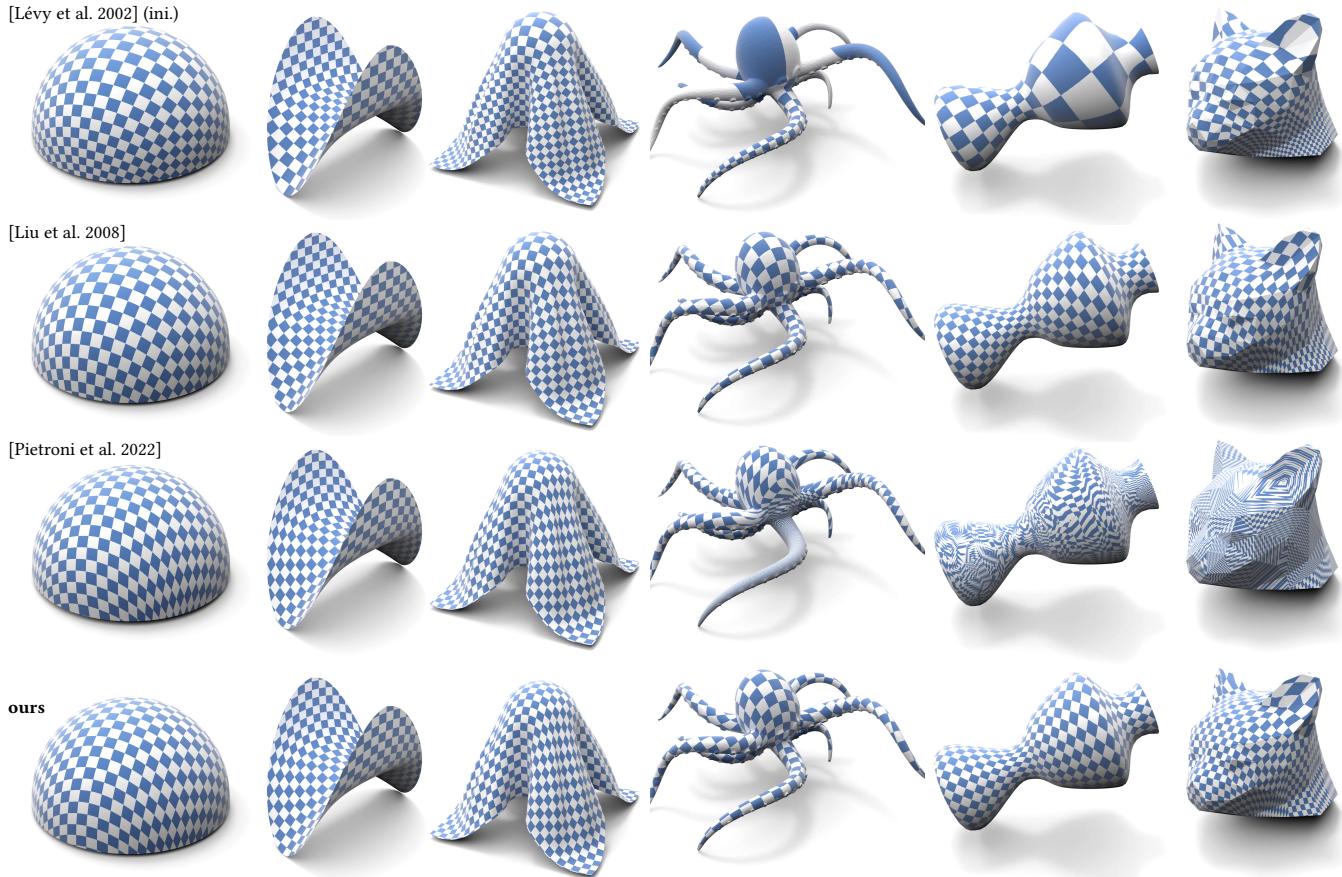


Fig. 9. Flattening disk-topology shapes using different parameterization methods. Note that [Pietroni et al. 2022] produces reasonable parameterizations in the first three examples but crashes in the last three; therefore we show the last iteration before it crashes. Shape complexity, Chebyshev error, and runtime are reported in the supplemental, Table 1. Note that we cut the octopus and the vase mesh into disk topology.

5.1 Chebyshev parameterization

To demonstrate that Chebyshev parameterization is a better choice than isometric parameterization for shapes that are realized with woven fabric, we conduct a physical experiment: we *tightly* wrap a piece of woven fabric around a 3D surface and mark its boundary using chalk. As shown in Fig. 8, we use wool and silk to wrap a hemisphere and a pringle shape, respectively. The chalk markings on the flattened fabric (despite some minor inaccuracies) reveal the ground-truth *uv*-mapping. It is evident that the Chebyshev flattening, computed using our algorithm, is much closer to the ground-truth than the isometric flattening computed by [Liu et al. 2008]. We conduct a similar experiment with more complicated, synthetic data using Blender [2024] in Fig. 3.

In Fig. 9 we show different parameterization results using least-squares conformal maps [Lévy et al. 2002], ARAP [Liu et al. 2008], YAF (short for “yarn-aware flattening”) [Pietroni et al. 2022], and our method. YAF attempts to compute a Chebyshev parameterization as part of their sewing pattern computation. It updates the *uv*-coordinates of a disk-topology patch to reduce the stretch along the yarn directions during its flattening iterations. YAF is quite efficient and works well on simple meshes. Its main limitation is the lack

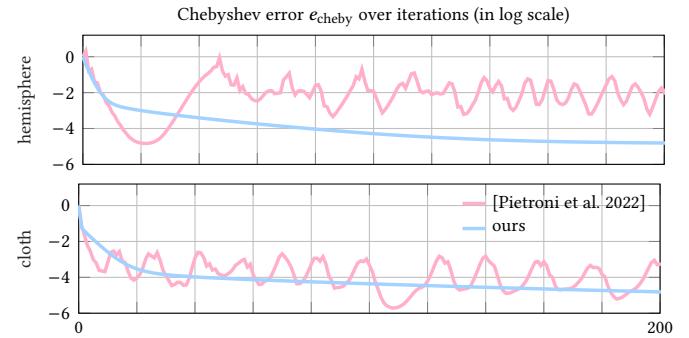


Fig. 10. Convergence behavior of [Pietroni et al. 2022] and our algorithm on two examples from Fig. 9. Chebyshev error is evaluated using Eq. (12).

of a principled mathematical discretization of a well-defined, continuous energy – YAF is more of an engineering approach to limit the stretch along the *uv*-axes in each iteration. As a consequence, there is no guarantee of convergence, as seen in its fluctuating behavior in Fig. 10. This also leads to heavy dependence on meshing quality and can cause YAF to be numerically unstable: For example, it fails



Fig. 11. The algorithm in [Pietroni et al. 2022] is numerically unstable: errors appearing in the early iterations accumulate and eventually lead to a crash.

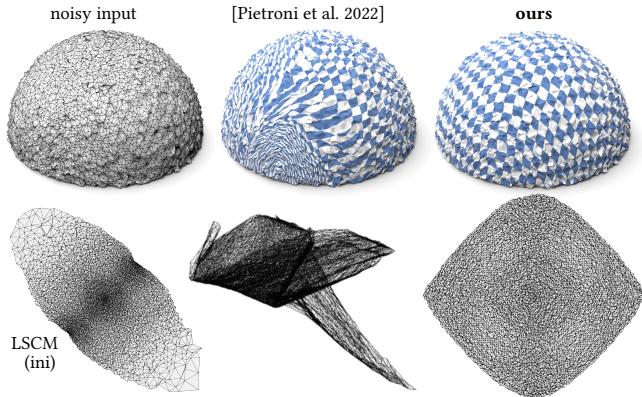


Fig. 12. Top: noisy hemisphere, its parameterization by [Pietroni et al. 2022] and our method. Bottom: the computed uv -domain, initialized by LSCM. While [Pietroni et al. 2022] can produce good results for a noise-free hemisphere (Fig. 9), it fails on a noisy one. Our method is more robust to noise and converges to a result similar to the noise-free setting shown in Fig. 8.

to produce any results for the two examples shown in Fig. 4; some faces collapse to a line during the flattening process, leading to a degenerated linear system. The example shown in Fig. 11 is another failure case of YAF, where we show its result one iteration before it crashes. We observe significant shearing and length distortion occurring in the ear region in early iterations, and YAF fails to recover from this scenario. Another limitation of YAF is its dependence on the isometric regularizer; the algorithm does not work without it. The associated weights in the objective function require fine-tuning for different examples.

In comparison, our method is based on a continuous energy that measures the deviation of the parameterization from a Chebyshev net, and we discretize it using the tried-and-true piecewise-linear approach on triangle meshes. Each iteration of our algorithm is designed to minimize the discrete energy and is guaranteed to not increase it, leading to robust convergence. Note that for arbitrary surfaces with large total absolute Gauss curvature (e.g. the last three examples shown in Fig. 9), a global Chebyshev net may not exist and some stretching distortion is to be expected. In these complex scenarios, our method successfully converges to results with significantly lower Chebyshev errors compared to YAF (see Table 1). Importantly, our Chebyshev flattening operates effectively without the need for regularizers or parameter tuning. Furthermore, it demonstrates robustness with respect to noisy inputs (see Fig. 12) and mesh discretization (see Fig. 13). Additionally, our formulation

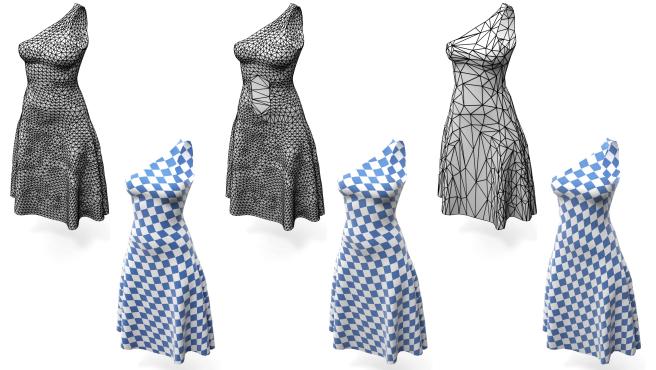


Fig. 13. Chebyshev parameterization computed using our algorithm on the same dress with different mesh discretizations. Our algorithm is more robust w.r.t. discretization and mesh resolution, compared to [Pietroni et al. 2022], which unfortunately failed on all these meshes.

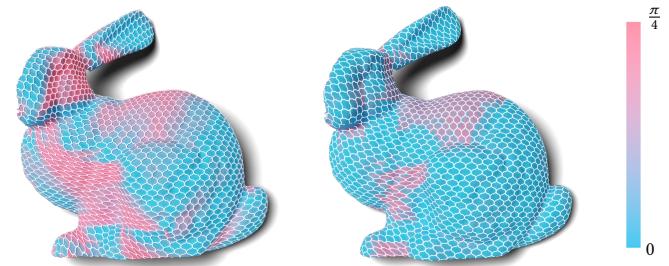


Fig. 14. Visualization of shearing angles overlaid with the net structure extracted from Chebyshev parameterization, comparing the results with (right) and without (left) shearing angle limitation (the limit is set to $\pi/4$).

allows precomputing the factorization of the system matrix in the global step and reusing it in every iteration, while YAF has to refactor its changing system matrix in each iteration.

5.2 Chebyshev net construction

Having a Chebyshev parameterization, we can easily construct a Chebyshev net to approximate the input 3D surface by extracting the uv -isolines. See Fig. 4 for some examples. Note that large shearing angles in the Chebyshev parameterization, defined as $|\frac{\pi}{2} - \alpha|$, where α is the yarn angle, can pose difficulties during physical fabrication processes [Liu et al. 2020].

To address this fabrication constraint, we propose enforcing the yarn angle α to lie within the range of $[\alpha_0, \pi - \alpha_0]$, which limits the amount of shearing in both diagonal directions to a user-specified threshold $\alpha_0 \in [0, \frac{\pi}{2}]$. This adjustment results in a refined search space for the Chebyshev parameterization, $\mathcal{B}' \subset \mathcal{B}$, which includes only Chebyshev-Jacobians stemming from a yarn angle in the allowed range $[\alpha_0, \pi - \alpha_0]$. This can be formally expressed as:

$$\mathcal{B}' = \left\{ Y \in \mathcal{B} \mid \det Y \leq \frac{1}{\sin \alpha_0} \right\}, \quad (13)$$

With our SVD-based formulation, we can easily ensure that the yarn angle lies within the allowed range $[\alpha_0, \pi - \alpha_0]$, or equally speaking,

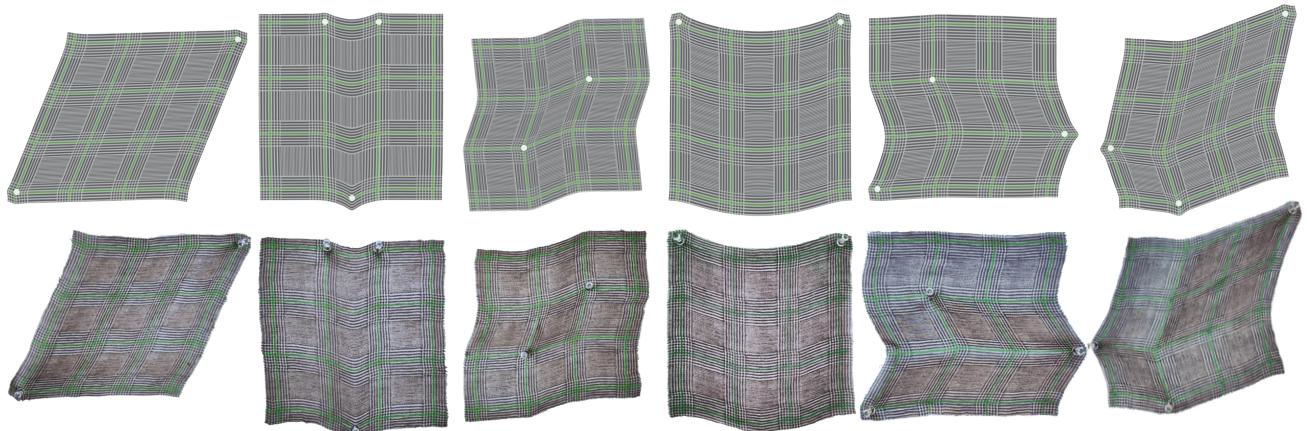


Fig. 15. Top: different deformations modeled by Chebyshev net deformation, constraints marked via white circles. Bottom: photographs of a piece of square silk fabric undergoing similar deformations.



Fig. 16. Starting with a cylinder mesh, we deform it by fixing its boundary vertices in different locations. We present four examples, each showcasing two types of visualizations: the texture mapping and an overlay of the shearing angle visualization with the extracted net structure. See Sec. 5.3 for details.

the solved Chebyshev-Jacobian Y lies in the refined search space \mathcal{B}' . Specifically, as discussed in Sec. 4, we know that $\forall Y \in \mathcal{B}$, the two eigenvalues of Y are $\lambda_1 = (\sqrt{1 + \cos \alpha})^{-1}$, $\lambda_2 = (\sqrt{1 - \cos \alpha})^{-1}$. In the local step of projecting Σ_t , we check whether the computed yarn angle is greater than α_0 ; if not, we set the singular values to the closest allowed values, namely $\lambda_1 = (\sqrt{1 + \cos \alpha_0})^{-1}$, $\lambda_2 = (\sqrt{1 - \cos \alpha_0})^{-1}$. Then, we have $\det Y = \lambda_1 \lambda_2 = 1/\sin \alpha_0 \in \mathcal{B}'$, i.e., the resulting Y is guaranteed to have a yarn angle within the allowed range. We can then proceed with the global step as usual.

Fig. 14 shows an example where we add shearing limit with $\alpha_0 = \pi/4$. Note that during the local step, we can ensure that the closest inverse Chebyshev Jacobians are drawn from \mathcal{B}' . However, there is no guarantee that the updated Jacobians after the global step still lie in \mathcal{B}' . This is why shearing angles greater than $\pi/4$ do not completely vanish in Fig. 14.

5.3 Chebyshev fabric manipulation

Existing cloth simulators that account for the anisotropic behavior of fabric often require the yarn directions to be provided as input. Our Chebyshev flattening can not only estimate the yarn directions for downstream cloth simulators, but also offers a straightforward solution for direct geometric manipulation of woven fabric. Given a 3D surface $\mathcal{M} = (\mathcal{V}, \mathcal{T})$, our goal is to determine its new deformed configuration $\mathcal{M}' = (\mathcal{V}', \mathcal{T})$ under external forces or by satisfying

specified constraints, while minimizing the fabric stretch along the parametric (yarn) lines. We first run our Chebyshev flattening algorithm to obtain $g : \mathcal{M} \rightarrow \mathbb{R}^2$. The image $\mathcal{U} = g(\mathcal{V})$ then gives the uv -domain of the surface and defines yarn directions (along the two parametric axes). The objective is to compute a deformation $f : \mathcal{U} \rightarrow \mathcal{M}'$, such that the Jacobian of f preserves the length along the parametric axes:

$$E(f) = \sum_{t \in \mathcal{T}} A_t \mathcal{D}(J_t, C)^2 = \sum_{t \in \mathcal{T}} A_t \min_{C \in \mathcal{C}} \|J_t - C\|_F. \quad (14)$$

Recall that C , defined in Eq. (3), is the space of Jacobians of Chebyshev nets. We can optimize the above energy using a local-global scheme. Note that in this case, the local step updating C simply projects J onto C by normalizing its two columns (using Eq. (2)).

Positional constraints. We can place positional constraints on a set of vertices of the input 3D mesh to guide its deformation while preserving its Chebyshev net structure as accurately as possible. Specifically, in the global step of minimizing Eq. (14), we can constrain certain vertices to specified positions from user input while solving for the updated vertex positions.

In Fig. 15, we deform a physical square piece of silk by fixing two or three points and compare the physical results with our deformation system. All the deformations exhibit noticeable shearing, which is unlikely to be accurately modeled by isometric methods

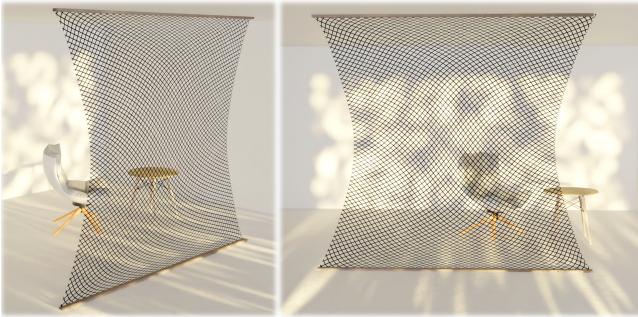


Fig. 17. We create a digital replica of the room divider made of a net, designed by Poli [2017], using our Chebyshev deformation tool.

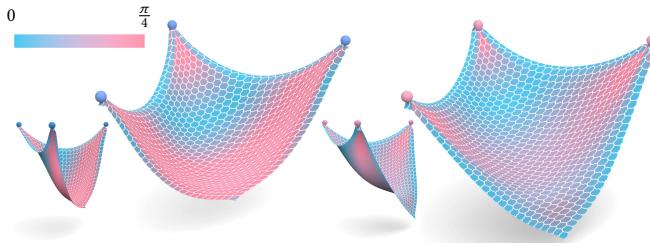


Fig. 18. We show an example of a hanging net, where we fix three vertices and add gravity, without (left) and with (right) shearing penalty.

that equally penalize stretch in all directions. We can see that our computed Chebyshev net-based deformation (Fig. 15, bottom) is faithful to the real-life deformations (Fig. 15, top). In Fig. 16 we show four examples of deforming a 3D topological cylinder mesh with different positional constraints, leading to interesting Chebyshev net configurations. The accompanying video shows real-time deformations. Note that the per-face Jacobians of the deformations are computed on the flattened, cut-open cylinder, while the 3D position updates are directly applied to vertices on the original cylinder, maintaining its topology. Fig. 17 shows an example of deforming a rectangle mesh.

Gravity and dynamics. We can incorporate gravity and dynamics into the deformation process, following ARAP-based dynamics [Jacobson et al. 2018]. Specifically, we introduce an additional step in the local-global optimization scheme for Eq. (14): updating the position of a vertex based on its current velocity, which is derived from the external forces, such as gravity, acting on each face. In the third column of Fig. 15, our result is obtained by adding gravity force to a square mesh with two points fixed. Fig. 18 presents an example where a square mesh is hung with three vertices fixed, allowing the fabric to deform under gravity. On the right we show the results with shearing penalty, where we add an extra regularizer to Eq. (14) that encourages the Jacobian to be closer to a rotation, i.e., $\sum_{t \in \mathcal{T}} \min_{R \in SO(2)} \|J_t - R\|_F$. This provides a soft shearing treatment compared to the hard shearing limit imposed by projection, as discussed in Sec. 5.2. Since the shearing regularization works purely on a face-level, essentially concerning intrinsic properties alone, it does not directly control the dihedral angles, or bending. We thus add a standard bi-Laplacian term to resolve the uncontrolled



Fig. 19. **Interactive draping.** Starting with a flat rectangular mesh representing a piece of woven fabric, our algorithm enables users to interactively drape it over a mannequin. The user’s manipulations are highlighted in red.

bending. The weights of the shearing and bending regularizations depend on the material to be modeled and are chosen empirically.

Interactive draping. One interesting application for Chebyshev deformation modeling is interactive digital draping, see Fig. 19 and the accompanying video. In this scenario, users can wrap a piece of “digital fabric” around, or pin part of the fabric onto a digital mannequin, much like designers do in real life. We prototype rudimentary collision avoidance, where a fabric vertex is pushed out with a certain strength once it is detected to be within the collider mesh. Fig. 19 shows an example where we wrap a rectangular fabric mesh around a mannequin to design a dress.

5.4 Implementation

We implemented our algorithm in C++. All experiments are conducted on a machine with an AMD Ryzen Threadripper 1950X 16-Core CPU. The full implementation can be found here: <https://github.com/oehria/woven-fabric-chebyshev>. Our Chebyshev flattening is parameter-free and efficient with robust convergence.

Initialization & stopping criterion. Since our algorithm works on topological disks, we introduce cuts to the input meshes when needed. All methods, including ARAP [Liu et al. 2008], YAF [Pietroni et al. 2022] and our method, are initialized by least-squares conformal parameterization (LSCM) [Lévy et al. 2002], where two vertices are selected and fixed in the uv domain with a distance equal to their geodesic distance on the input surface. Our Chebyshev flattening algorithm is parameter-free. We terminate the iterations when the normalized change of the uv -coordinates is smaller than 10^{-4} . For the comparisons to YAF, we use the released code from the authors¹. When crashes happen in their code (some faces collapse to a line during the flattening process), we save the result from the last iteration before the crash.

Robustness w.r.t. initializations. Our method is robust to noisy input meshes, see examples in Figures 12 and 13. For different initialization strategies, we first experiment with three different types of initialization for our algorithm, namely harmonic mapping with the boundary vertices fixed to a circle, LSCM, and ARAP. Fig. 20

¹<https://github.com/nicopietroni/parafashion>

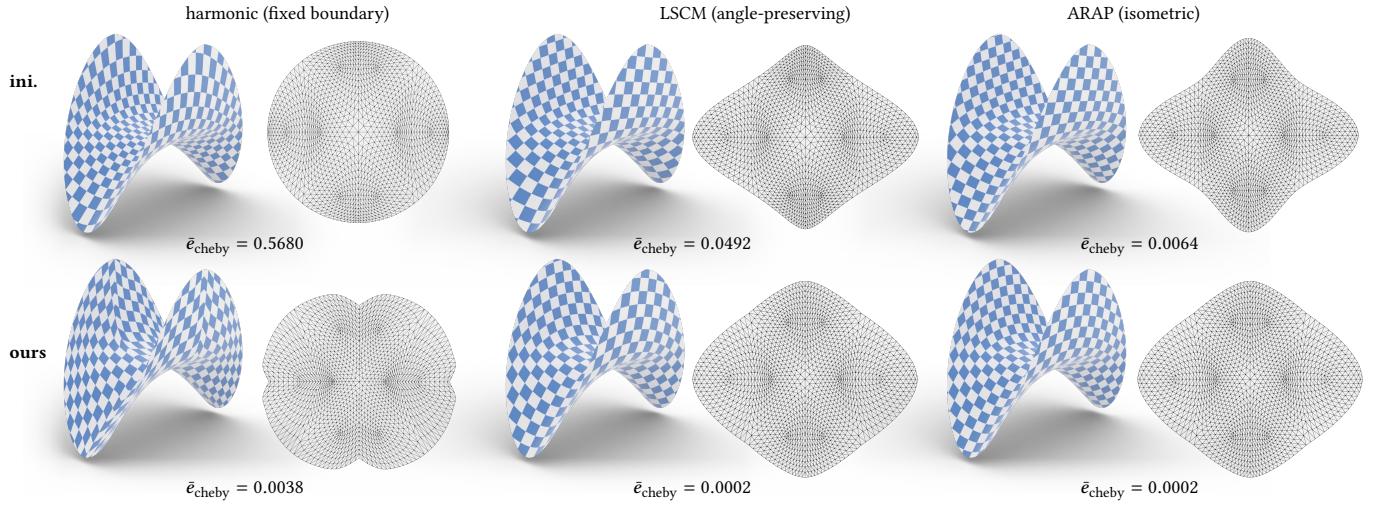


Fig. 20. We run our Chebyshev flattening method starting from three different initializations, shown in the top row, with the corresponding results shown on the bottom row: harmonic parameterization with the boundary vertices fixed to a circle (left), LSCM [Lévy et al. 2002] (middle), and ARAP [Liu et al. 2008] (right), initialized by LSCM. For each result, we show the computed flattening, the texture mapping, and the Chebyshev error normalized by the surface area.

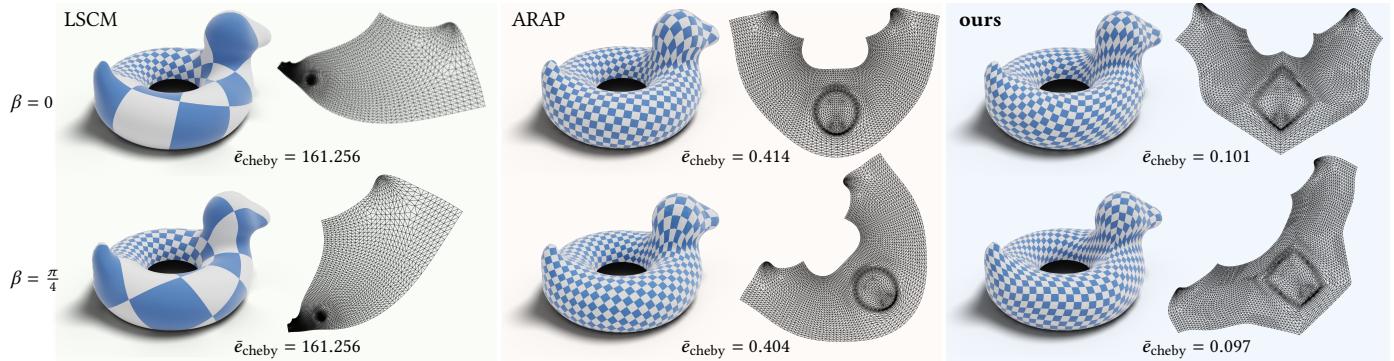


Fig. 21. We initialize the parameterization algorithms with LSCM [Lévy et al. 2002] (top row) and its rotation by 45° (bottom row) and show the resulting converged ARAP parameterization [Liu et al. 2008] and our Chebyshev parameterization. For each result, we show the textured 3D mesh and its corresponding uv space and report the average Chebyshev error.

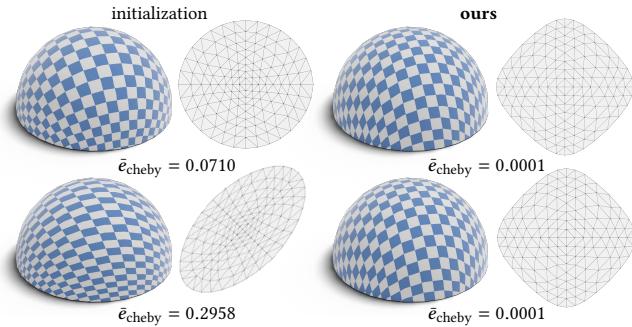


Fig. 22. We run our algorithm with the normal LSCM [Lévy et al. 2002] initialization (top) and a distorted version (bottom). Both runs converge to the same optimal solution for the hemisphere.

shows that our method converges to the same parameterization with LSCM and ARAP initialization. When starting from the harmonic mapping, our method converges to a slightly worse local minimum. We generally observe that for many shapes, especially the more complex ones, different initializations yield slightly different results, but with similarly low energies. This is due to the very non-convex nature of the energy. One such example is shown in Fig. 21, where a 45°-rotated LSCM initialization yields visibly different results to the original, but their energies are not too different. Additionally, it can be seen that the energies are significantly lower than the ones for the LSCM initializations and the converged ARAP results. For simpler examples such as the hemisphere, Fig. 22 shows that distorting the initialization does not affect the end result, since there is a much more distinct optimal solution.

Table 1. For shapes in Fig. 9: we define convergence for all approaches when the norm of the relative change in uv s is below $1e - 4$, with maximum number of iterations set to 1000. For YAF [Pietroni et al. 2022] we chose stretching penalty weight of 30 and rigid regularization weight of 1. In the convergence column, * marks the two examples whose convergence behaviour was more closely shown in Fig. 10, and $\dagger j$ denotes that the example fully diverged, producing NaN values after j iterations, in which case we make our measurements on the last iteration before the NaN values occurred. The runtime corresponding to a run without convergence is marked with italics. LSCM [Lévy et al. 2002] is a linear method, so convergence issues do not apply.

shape	method	complexity		Chebyshev error $\mathcal{D}(B_t^{-1}, C)^2$ (see Eq. (12))			runtime (sec.)	converged?
		#faces	#vtx	min.	max.	avg.		
hemisphere	conformal [Lévy et al. 2002]	16368	8313	1.42e-07	3.49e-01	7.28e-02	0.243	N/A
	isometric [Liu et al. 2008]			4.31e-05	7.52e-02	2.69e-02	0.129	yes
	Chebyshev [Pietroni et al. 2022]			1.66e-11	1.75e-04	2.43e-06	1.550	yes*
	Chebyshev (ours)			3.50e-08	1.03e-03	2.64e-04	0.346	yes*
pringle	conformal [Lévy et al. 2002]	3072	1601	1.41e-05	3.37e-01	4.92e-02	0.023	N/A
	isometric [Liu et al. 2008]			1.32e-05	5.76e-02	6.50e-03	0.016	yes
	Chebyshev [Pietroni et al. 2022]			6.75e-13	4.14e-04	3.86e-05	0.081	yes
	Chebyshev (ours)			3.03e-09	1.23e-03	5.91e-05	0.113	yes
cloth	conformal [Lévy et al. 2002]	3872	2025	2.03e-05	4.72e-01	3.39e-01	0.035	N/A
	isometric [Liu et al. 2008]			2.28e-07	2.08e-01	1.95e-02	0.022	yes
	Chebyshev [Pietroni et al. 2022]			1.54e-10	3.88e-05	1.98e-06	14.90	no*
	Chebyshev (ours)			4.09e-10	1.03e-03	5.10e-05	0.085	yes*
octopus	conformal [Lévy et al. 2002]	15141	26968	4.92e-08	2.83e+12	1.46e+08	0.229	N/A
	isometric [Liu et al. 2008]			3.93e-08	4.95e+03	1.64e-01	0.320	yes
	Chebyshev [Pietroni et al. 2022]			3.39e-08	9.89e+07	1.55e+03	0.242	no ^{†1}
	Chebyshev (ours)			1.39e-10	1.05e+05	3.84e+00	3.010	yes
vase	conformal [Lévy et al. 2002]	1536	833	1.53e-02	4.51e+01	1.42e+01	0.010	N/A
	isometric [Liu et al. 2008]			5.51e-05	9.70e+00	8.00e-02	0.007	yes
	Chebyshev [Pietroni et al. 2022]			5.29e-04	9.47e+06	2.09e+05	0.080	no ^{†15}
	Chebyshev (ours)			1.01e-06	4.34e-01	9.83e-03	0.212	yes
cat	conformal [Lévy et al. 2002]	248	131	9.51e-03	1.02e+03	1.45e+01	0.003	N/A
	isometric [Liu et al. 2008]			6.59e-03	5.16e+00	7.45e-01	0.030	yes
	Chebyshev [Pietroni et al. 2022]			8.89e-03	1.69e+08	6.72e+05	0.068	no ^{†84}
	Chebyshev (ours)			2.21e-05	8.74e+01	8.07e-01	6.770	yes



Fig. 23. Chebyshev parameterization on a wavy plane example. The convergence is reported in Fig. 24

Runtime and convergence. Our local-global approach is quite efficient: for a mesh of 20K faces, it takes about 2 seconds to converge. Table 1 reports the mesh complexity and runtime for the shapes in Fig. 9. Our algorithm is designed to not increase the energy in each iteration, resulting in robust convergence. Figures 10 and 24 report the Chebyshev error over iterations for YAF and our method. YAF yields good results on simple shapes. However, there is no guarantee of convergence even on some simple shapes with regular triangulation, such as the hemisphere example shown in Fig. 9.

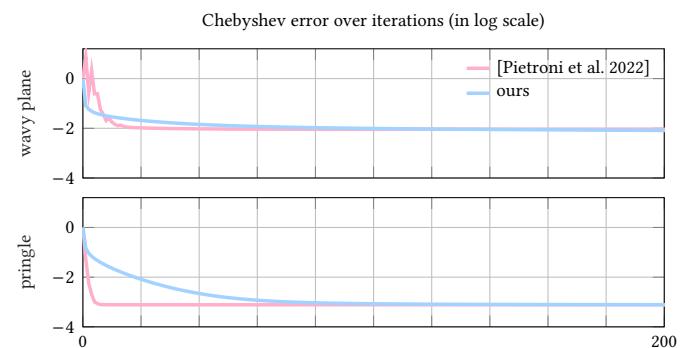


Fig. 24. Convergence behavior of [Pietroni et al. 2022] and our algorithm on two examples. The Chebyshev error is evaluated using Eq. (12).

6 CONCLUSION, LIMITATIONS & FUTURE WORK

In this work, we highlight the advantages of anisotropic mesh parameterization based on Chebyshev nets when the parameterized surface is meant to represent draped woven fabric or net material. In

such case, our proposed Chebyshev deformation energy is a more accurate objective than isotropic, isometry-based distortion measures. We propose a novel efficient algorithm for computing Chebyshev parameterizations based on the singular value decomposition of the Jacobian matrices. Additionally, we showcase the effectiveness of our algorithm in fitting woven fabric to 3D surfaces, extracting Chebyshev nets for surface approximation, and facilitating interactive digital fabric manipulation.

Our method also has some limitations. Currently, we use a Taylor expansion to approximate the singular values in the local step, which could be suboptimal. An interesting direction for future work is to develop a better strategy for the local step with theoretical guarantees. Our Chebyshev parameterization algorithm currently only supports disk-topology meshes, as it is specifically designed for modeling woven fabric, which naturally has disk topology without singularities. In the future, it would be interesting to generalize our SVD-based formulation to field-based parameterization and handle singularities, which can be beneficial for other applications, such as wire net design. Our interactive woven fabric manipulation algorithm is purely geometric and does not handle self-collisions, nor does it properly model specific material properties. It would be interesting to integrate it with more advanced simulators to improve efficiency and realism.

ACKNOWLEDGMENTS

This work was supported in part by the ERC Consolidator Grant No. 101003104 (MYCLOTH). The authors would like to thank the anonymous reviewers for their valuable feedback. Special thanks to Corentin Dumery and Nico Pietroni for their help with the comparison to YAF, and to all members of IGL for the insightful discussions and kind support.

REFERENCES

- Masaki Aono, David Breen, and Michael Wozny. 2001. Modeling Methods for the Design of 3D Broadcloth Composite Parts. *Computer Aided Design* 33 (11 2001), 989–1007. [https://doi.org/10.1016/S0010-4485\(00\)00135-4](https://doi.org/10.1016/S0010-4485(00)00135-4)
- Masaki Aono, David E Breen, and Michael J Wozny. 1994. Fitting a woven-cloth model to a curved surface: mapping algorithms. *Computer Aided Design* 26, 4 (1994), 278–292.
- 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.
- David Baraff and Andrew Witkin. 1998. Large Steps in Cloth Simulation. In *Proc. SIGGRAPH*. 43–54.
- Blender. 2024. Blender. <https://docs.blender.org/manual/en/latest/physics/cloth/index.html>
- David E Breen. 2000. A survey of cloth modeling methods. In *Cloth modeling and animation*. AK Peters/CRC Press, 19–53.
- David E Breen, Donald H House, and Michael J Wozny. 1994. Predicting the drape of woven cloth using interacting particles. (1994), 365–372.
- Pafnuty Lvovich Chebyshev. 1878. Sur la coupe des vêtements, "On the cutting of garments". In *Association française pour l'avancement des sciences, Congrès de Paris*. 154–155.
- Kwang-Jin Choi and Hyeong-Seok Ko. 2002. Stable but Responsive Cloth. *ACM Trans. Graph.* 21, 3 (July 2002), 604–611.
- Gabriel Cirio, Jorge Lopez-Moreno, David Miraut, and Miguel A Otaduy. 2014. Yarn-level simulation of woven cloth. *ACM Trans. Graph.* 33, 6 (2014), 207:1–207:11.
- CLO Virtual Fashion. 2022. Clo3D. <https://clo3d.com/en/>
- Philippe Decaudin, Dan Julius, Jamie Withner, Laurence Boissieux, Alla Sheffer, and Marie-Paule Cani. 2006. Virtual Garments: A Fully Geometric Approach for Clothing Design. *Computer Graphics Forum* 25, 3 (2006), 1–10.
- Michael S Floater and Kai Hormann. 2005. Surface parameterization: a tutorial and survey. *Advances in multiresolution for geometric modelling* (2005), 157–186.
- 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 (2014).
- Rony Goldenthal, David Harmon, Raanan Fattal, Michel Bercovier, and Eitan Grinspun. 2007. Efficient simulation of inextensible cloth. *ACM Trans. Graph.* 26, 3 (jul 2007), 49–es. <https://doi.org/10.1145/1276377.1276438>
- Kai Hormann, Bruno Lévy, and Alla Sheffer. 2007. Mesh parameterization: Theory and practice. (2007).
- Yuki Igarashi, Takeo Igarashi, and Hiromasa Suzuki. 2009. Interactive Cover Design Considering Physical Constraints. *Computer Graphics Forum* 28 (10 2009), 1965–1973. <https://doi.org/10.1111/j.1467-8659.2009.01575.x>
- Alec Jacobson, Daniele Panozzo, et al. 2018. libigl: A simple C++ geometry processing library. <https://libigl.github.io/>.
- Dan Julius, Vladislav Kraevoy, and Alla Sheffer. 2005. D-Charts: Quasi-Developable Mesh Segmentation. *Computer Graphics Forum* 24, 3 (2005), 581–590. <https://doi.org/10.1111/j.1467-8659.2005.00883.x>
- Jonathan M Kaldor, Doug L James, and Steve Marschner. 2010. Efficient yarn-based cloth with adaptive contact linearization. *ACM Trans. Graph.* 29, 4 (2010), 105:1–105:10.
- Jan Koenderink and Andrea van Doorn. 1998. Shape from Chebyshev nets. In *European Conference on Computer Vision (ECCV)*. Springer, 215–225.
- Mina Konaković, Keenan Crane, Bailin Deng, Sofien Bouaziz, Daniel Piker, and Mark Pauly. 2016. Beyond developable: computational design and fabrication with auxetic materials. *ACM Trans. Graph.* 35, 4, Article 89 (jul 2016), 11 pages. <https://doi.org/10.1145/2897824.2925944>
- Mina Konaković-Luković, Julian Panetta, Keenan Crane, and Mark Pauly. 2018. Rapid deployment of curved surfaces via programmable auxetics. *ACM Trans. Graph.* 37, 4 (2018), 1–13.
- Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérôme Maillot. 2002. Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph.* 21, 3 (jul 2002), 362–371. <https://doi.org/10.1145/566654.566590>
- Minchen Li, Danny M. Kaufman, Vladimir G. Kim, Justin Solomon, and Alla Sheffer. 2018. OptCuts: joint optimization of surface cuts and parameterization. *ACM Trans. Graph.* 37, 6, Article 247 (dec 2018), 13 pages. <https://doi.org/10.1145/3272127.3275042>
- Rui-Zeng Li, Jia-Peng Guo, Qi Wang, Shuangming Chai, Ligang Liu, and Xiao-Ming Fu. 2022. Interactive Editing of Discrete Chebyshev Nets. *Computer Graphics Forum* (2022). <https://doi.org/10.1111/cgf.14462>
- Hao-Yu Liu, Zhong-Yuan Liu, Zheng-Yu Zhao, Ligang Liu, and Xiao-Ming Fu. 2020. Practical fabrication of discrete chebyshev nets. *Computer Graphics Forum* 39, 7 (2020), 13–26.
- Ligang Liu, Lei Zhang, Yin Xu, Craig Gotsman, and Steven J Gortler. 2008. A local/global approach to mesh parameterization. *Computer Graphics Forum* 27, 5 (2008), 1495–1504.
- Tiantian Liu, Adam W. Bargteil, James F. O'Brien, and Ladislav Kavan. 2013. Fast Simulation of Mass-Spring Systems. *ACM Trans. Graph.* 32, 6 (Nov. 2013), 209:1–7.
- C Mack and HM Taylor. 1956. The fitting of woven cloth to surfaces. *Journal of the Textile Institute Transactions* 47, 9 (1956), T477–T488.
- Yannick Masson and Laurent Monasse. 2017. Existence of global Chebyshev nets on surfaces of absolute Gaussian curvature less than 2π . *Journal of Geometry* 108, 1 (2017), 25–32.
- James McCartney, BK Hinds, and KW Chong. 2005. Pattern flattening for orthotropic materials. *Computer-Aided Design* 37, 6 (2005), 631–644.
- James McCartney, BK Hinds, and BL Seow. 1999. The flattening of triangulated surfaces incorporating darts and gussets. *Computer Aided Design* 31, 4 (1999), 249–260.
- James McCartney, BK Hinds, BL Seow, and D Gong. 2000. An energy based model for the flattening of woven fabrics. *Journal of Materials Processing Technology* 107, 1–3 (2000), 312–318.
- Nicolas Montagne, Cyril Doutre, Xavier Tellier, Corentin Fivet, and Olivier Baverel. 2020. Voss surfaces: A design space for geodesic gridshells. *Journal of the International Association for Shell and Spatial Structures* 61, 4 (2020), 255–263.
- Yuki Mori and Takeo Igarashi. 2007. Plushie: an interactive design system for plush toys. *ACM Trans. Graph.* 26, 3 (2007), 45–es.
- Rahul Narain, Armin Samii, and James F. O'Brien. 2012. Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. Graph.* 31, 6, Article 152 (nov 2012), 10 pages. <https://doi.org/10.1145/2366145.2366171>
- Xiongqi Peng, Zaoyang Guo, Tongliang Du, and Woong-Ryeol Yu. 2013. A simple anisotropic hyperelastic constitutive model for textile fabrics with application to forming simulation. *Composites Part B: Engineering* 52 (2013), 275–281.
- Nico Pietroni, Corentin Dumery, Raphaël Falque, Mark Liu, Teresa Vidal-Calleja, and Olga Sorkine-Hornung. 2022. Computational Pattern Making from 3D Garment Models. *ACM Trans. Graph.* 41, 4 (2022), 157:1–14.
- Franco Poli. 2017. <https://www.matteograssi.com/en/products/loose-furniture-furnishing-elements/t-net>
- Roi Poranne, Marco Tarini, Sandro Huber, Daniele Panozzo, and Olga Sorkine-Hornung. 2017. Autocuts: Simultaneous Distortion and Cut Optimization for UV Mapping. *ACM Trans. Graph.* 36, 6 (2017).
- Michael Rabinovich, Tim Hoffmann, and Olga Sorkine-Hornung. 2018. Discrete Geodesic Nets for Modeling Developable Surfaces. *ACM Trans. Graph.* 37, 2 (2018).

- Rajcoomar B Ramgulam. 2001. A novel algorithm for fitting a woven-cloth to complex surfaces. *International Journal of Clothing Science and Technology* 13, 3/4 (2001), 198–207.
- Richard E Robertson, ES Hsieh, EN Sickafus, and GSY Yeh. 1981. Fiber rearrangements during the molding of continuous fiber composites. I. Flat cloth to a hemisphere. *Polymer composites* 2, 3 (1981), 126–131.
- Richard E Robertson, ES Hsieh, and GSY Yeh. 1984. Continuous fiber rearrangements during the molding of fiber composites. II. Flat cloth to a rounded cone. *Polymer Composites* 5, 3 (1984), 191–197.
- Kenneth Rose, Alla Sheffer, Jamie Wither, Marie-Paule Cani, and Boris Thibert. 2007. Developable surfaces from arbitrary sketched boundaries. In *Proc. SGP*. Eurographics Association, 163–172.
- Andrew O Sageman-Furnas, Albert Chern, Mirela Ben-Chen, and Amir Vaxman. 2019. Chebyshev nets from commuting polyvector fields. *ACM Trans. Graph.* 38, 6 (2019), 1–16.
- Pedro V. Sander, John Snyder, Steven J. Gortler, and Hugues Hoppe. 2001. Texture mapping progressive meshes. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. Association for Computing Machinery, New York, NY, USA, 409–416. <https://doi.org/10.1145/383259.383307>
- Rohan Sawhney and Keenan Crane. 2017. Boundary first flattening. *ACM Trans. Graph.* 37, 1 (2017), 1–14.
- Alla Sheffer, Bruno Lévy, Maxim Mogilnitsky, and Alexander Bogomyakov. 2005. ABF++: fast and robust angle based flattening. *ACM Trans. Graph.* 24, 2 (2005), 311–330.
- Alla Sheffer, Emil Praun, Kenneth Rose, et al. 2007. Mesh parameterization methods and their applications. *Foundations and Trends® in Computer Graphics and Vision* 2, 2 (2007), 105–171.
- Olga Sorkine and Marc Alexa. 2007. As-rigid-as-possible surface modeling. In *Proc. SGP (SGP '07)*. Eurographics Association, 109–116.
- Olga Sorkine, Daniel Cohen-Or, Rony Goldenthal, and Dani Lischinski. 2002. Bounded-distortion piecewise mesh parameterization. In *Proceedings of IEEE Visualization*. IEEE Computer Society, 355–362.
- François Trochu, A Hammami, and Y Benoit. 1996. Prediction of fibre orientation and net shape definition of complex composite parts. *Composites Part A: Applied Science and Manufacturing* 27, 4 (1996), 319–328.
- Betty P van West, RB Pipes, and M Keefe. 1990. A simulation of the draping of bidirectional fabrics over arbitrary surfaces. *Journal of the textile Institute* 81, 4 (1990), 448–460.
- Charlie C.L. Wang. 2007. WireWarping: A fast surface flattening approach with length-preserved feature curves. *Computer Aided Design* 40, 3 (3 Dec. 2007), 381–395. <https://doi.org/10.1016/j.cad.2007.11.011>
- Charlie C.L. Wang, Kai Tang, and Benjamin ML Yeung. 2005. Freeform surface flattening based on fitting a woven mesh model. *Computer Aided Design* 37, 8 (2005), 799–814.
- Huamin Wang, James F. O'Brien, and Ravi Ramamoorthi. 2011. Data-Driven Elastic Models for Cloth: Modeling and Measurement. *ACM Trans. Graph.* 30, 4, Article 71 (jul 2011), 12 pages. <https://doi.org/10.1145/2010324.1964966>
- Xiaoting Zhang, Guoxin Fang, Melina Skouras, Gwenda Gieseler, Charlie C.L. Wang, and Emily Whiting. 2019. Computational Design of Fabric Formwork. *ACM Trans. Graph.* 38, 4 (2019), 109:1–109:13.

A PROOF FOR SVD OF CHEBYSHEV NET'S JACOBIAN

Let J be the 2×2 Jacobian of an orientation preserving Chebyshev net, as defined in Sec. 3 of the paper, and (\mathbf{u}, \mathbf{v}) the two orthonormal frame vectors in the uv -domain, $\mathbf{u} = (1, 0)$, $\mathbf{v} = (0, 1)$. Since a Chebyshev net by definition preserves the lengths along the frame axes, while at the same time allowing the change of angles by shearing, we have

$$\|J\mathbf{u}\|_2^2 = \|J\mathbf{v}\|_2^2 = 1. \quad (15)$$

We investigate what the above constraints imply about the singular value decomposition $J = U\Sigma V^\top$, where U and V are 2D rotation matrices and Σ is a diagonal matrix with singular values σ_1, σ_2 (U and V can be chosen to be rotations since $\det J > 0$). Since U is orthogonal, we know $\|U\mathbf{x}\|_2 = \|\mathbf{x}\|_2, \forall \mathbf{x}$. Therefore, we have

$$1 = \|J\mathbf{u}\|_2^2 = \|U\Sigma V^\top \mathbf{u}\|_2^2 = \|\Sigma V^\top \mathbf{u}\|_2^2, \quad (16a)$$

$$1 = \|J\mathbf{v}\|_2^2 = \|U\Sigma V^\top \mathbf{v}\|_2^2 = \|\Sigma V^\top \mathbf{v}\|_2^2. \quad (16b)$$

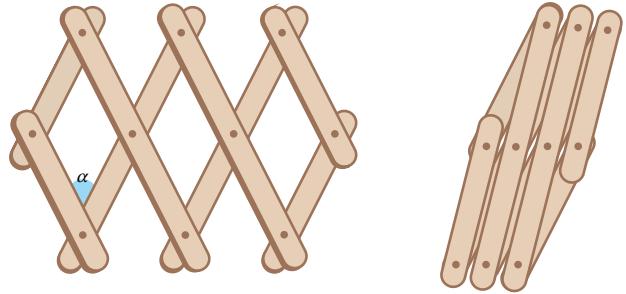


Fig. 25. An example of a wooden hanger using a Chebyshev net with opening (yarn) angle α , in expanded (left) and collapsed (right) state.

The singular value matrix Σ can be denoted as:

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}, \quad (17)$$

where $\sigma_1 \geq \sigma_2 > 0$. We assume both singular values in Σ to be positive since we assume the parameterization is regular. We further denote V as a 2D rotation matrix that rotates the 2D plane counterclockwise by angle γ , i.e., :

$$V = \begin{pmatrix} \cos \gamma & -\sin \gamma \\ \sin \gamma & \cos \gamma \end{pmatrix}. \quad (18)$$

Plugging Σ and V into Eq. (16a) we can obtain:

$$\begin{aligned} 1 &= \left\| \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} \begin{pmatrix} \cos \gamma & \sin \gamma \\ -\sin \gamma & \cos \gamma \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\|_2^2 = \left\| \begin{pmatrix} \sigma_1 \cos \gamma \\ -\sigma_2 \sin \gamma \end{pmatrix} \right\|_2^2 \\ &= \sigma_1^2 \cos^2 \gamma + \sigma_2^2 \sin^2 \gamma \\ &= (\sigma_1^2 - \sigma_2^2) \cos^2 \gamma + \sigma_2^2 (\cos^2 \gamma + \sin^2 \gamma) \\ &= (\sigma_1^2 - \sigma_2^2) \cos^2 \gamma + \sigma_2^2. \end{aligned} \quad (19)$$

Similarly, plugging Σ and V into Eq. (16b) and following a similar derivation, we can obtain:

$$(\sigma_1^2 - \sigma_2^2) \sin^2 \gamma + \sigma_2^2 = 1. \quad (20)$$

Subtracting Eq. (19) from Eq. (20) leads to:

$$(\sigma_1^2 - \sigma_2^2) (\sin^2 \gamma - \cos^2 \gamma) = 0. \quad (21)$$

- In case of $\sigma_1 \neq \sigma_2$, we have $\sin^2 \gamma = \cos^2 \gamma \Rightarrow \gamma = \pm \frac{\pi}{4}, \pm \frac{3\pi}{4}$.
- In case of $\sigma_1 = \sigma_2$, according to Eq. (16a), we have:

$$1 = \|J\mathbf{u}\|_2 = \|U\Sigma V^\top \mathbf{u}\|_2 = \sigma_1 \|UV^\top \mathbf{u}\|_2 = \sigma_1 \|\mathbf{u}\|_2 = \sigma_1, \quad (22)$$

i.e., $\sigma_1 = \sigma_2 = 1$, i.e., $J = UV^\top$ is a rotation by some angle θ . The SVD is not unique, and we can always choose V to be a rotation by $\pm \frac{\pi}{4}$ or $\pm \frac{3\pi}{4}$, and choose U to be a rotation by $\theta \mp \frac{\pi}{4}$, resp. $\theta \mp \frac{3\pi}{4}$.

Therefore, from Eq. (21) we can conclude that V is a rotation by angle $\pm \frac{\pi}{4}$ or $\pm \frac{3\pi}{4}$. Without loss of generality, we can simply assume V is a rotation by angle $\frac{\pi}{4}$, as the other solutions are equivalent SVD with permuted u, v axes and with flipped signs.

Summing Equations (19) and (20) together leads to:

$$\sigma_1^2 + \sigma_2^2 = 2. \quad (23)$$

We can also compute the yarn angle α , i.e., the angle between the vectors $J\mathbf{u}$ and $J\mathbf{v}$:

$$\begin{aligned}\langle J\mathbf{u}, J\mathbf{v} \rangle &= \|J\mathbf{u}\| \|J\mathbf{v}\| \cos \alpha = \cos \alpha \\ &= (U\Sigma V^T \mathbf{u})^T U\Sigma V^T \mathbf{v} = \mathbf{u}^T V\Sigma U^T U\Sigma V^T \mathbf{v} = \mathbf{u}^T V\Sigma^2 V^T \mathbf{v} \\ &= (1 \ 0) \begin{pmatrix} \cos \gamma & -\sin \gamma \\ \sin \gamma & \cos \gamma \end{pmatrix} \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix} \begin{pmatrix} \cos \gamma & \sin \gamma \\ -\sin \gamma & \cos \gamma \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= (\sigma_1^2 - \sigma_2^2) \sin \gamma \cos \gamma \\ &\Rightarrow \cos \alpha = \frac{1}{2} (\sigma_1^2 - \sigma_2^2). \end{aligned} \quad (24)$$

Subtracting and summing Equations (23) and (24), we have:

$$\sigma_1^2 = 1 + \cos \alpha, \quad \sigma_2^2 = 1 - \cos \alpha. \quad (25)$$

Since $\sigma_1 \geq \sigma_2 > 0$, we have $\sigma_1 = \sqrt{1 + \cos \alpha}$, $\sigma_2 = \sqrt{1 - \cos \alpha}$. We can compute the determinant of the Jacobian:

$$\begin{aligned}\det(J) &= \det(U\Sigma V^T) = \det(U) \det(\Sigma) \det(V) = \sigma_1 \sigma_2 \\ &= \sqrt{(1 + \cos \alpha)(1 - \cos \alpha)} = \sin \alpha.\end{aligned} \quad (26)$$

To summarize, applying SVD to the Jacobian of a Chebyshev net provides us with the following information:

- The V matrix can always be chosen to be a rotation by $\frac{\pi}{4}$.
- The two singular values always lie on a circle with radius $\sqrt{2}$, i.e., $\sigma_1^2 + \sigma_2^2 = 2$.
- The yarn angle α can be directly derived from the singular values, i.e., $\cos \alpha = \frac{1}{2} (\sigma_1^2 - \sigma_2^2)$.
- The singular values can also be directly derived from the yarn angle, i.e., $\sigma_1 = \sqrt{1 + \cos \alpha}$, $\sigma_2 = \sqrt{1 - \cos \alpha}$.
- The determinant of the Jacobian is equal to the sine of the yarn angle, i.e., $\det(J) = \sin \alpha$.