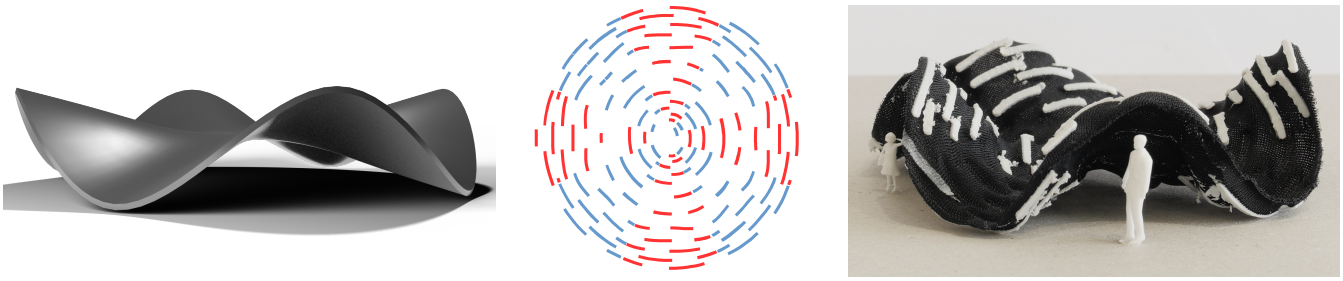
EUROGRAPHICS 2022 / R. Chaine and M. H. Kim
(Guest Editors)

Volume 41 (2022), Number 2

Computational Design of Self-Actuated Surfaces by Printing Plastic Ribbons on Stretched Fabric

David Jourdan[1] Mélina Skouras[2] Etienne Vouga[3] Adrien Bousseau[1]

[1]Inria, Université Côte d'Azur, France
[2]Université Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, France
[3]University of Texas at Austin, USA



(a) Target surface (b) Ribbon layout (c) Deployed surface

Figure 1: *Given a target 3D surface (a), our method computes a flat layout of plastic ribbons (b) which, when printed on pre-stretched fabric, causes the surface to deploy into 3D when the fabric is released (c). We designed a custom frame to easily print ribbons on both sides of the fabric (b, red and blue), allowing us to reproduce surfaces with positive and negative extrinsic curvature.*

Abstract

We introduce a new mechanism for self-actuating deployable structures, based on printing a dense pattern of closely-spaced plastic ribbons on sheets of pre-stretched elastic fabric. We leverage two shape-changing effects that occur when such an assembly is printed and allowed to relax: first, the incompressible plastic ribbons frustrate the contraction of the fabric back to its rest state, forcing residual strain in the fabric and creating intrinsic curvature. Second, the differential compression at the interface between the plastic and fabric layers yields a bilayer effect *in the direction of the ribbons, making each ribbon buckle into an arc at equilibrium state and creating extrinsic curvature. We describe an inverse design tool to fabricate low-cost, lightweight prototypes of freeform surfaces using the controllable directional distortion and curvature offered by this mechanism. The core of our method is a parameterization algorithm that bounds surface distortions along and across principal curvature directions, along with a pattern synthesis algorithm that covers a surface with ribbons to match the target distortions and curvature given by the aforementioned parameterization. We demonstrate the flexibility and accuracy of our method by fabricating and measuring a variety of surfaces, including nearly-developable surfaces as well as surfaces with positive and negative mean curvature, which we achieve thanks to a simple hardware setup that allows printing on both sides of the fabric.*

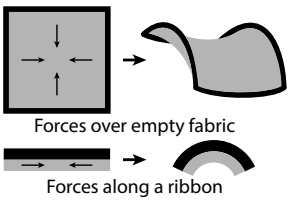
Keywords: fabrication, parameterization, inverse design, self-actuation, printing on fabric

1. Introduction

Deployable structures—structures that are fabricated flat and subsequently actuate into the third dimension—are a subject of intense interest in engineering, manufacturing, and architecture, as such structures can have high geometric complexity, while remaining accessible to traditional 2D manufacturing techniques, and retaining many of the advantages of flat-fabricated objects such as ease of transport and storage. Actuation from the flat to deformed state can be achieved either via applying external loads on the structure (pushing or pulling points of the surface) [KPCP18; PKI*19], or via *self-actuation* where changes in the internal stress of the assembly trigger shape changes [BvRL*19; GMN*16; GMB17; PIC*21].

This paper contributes to the latter family of methods: we explore a new mechanism for self-actuation, based on printing plastic ribbons on sheets of pre-stretched elastic fabric. Relative to other self-actuation strategies, this modality is simple and low-cost—requiring only a commodity 3D printer and a rigid frame for clamping the fabric. Since the fabric is pre-stretched and the plastic ribbons are not, the ensemble acts as an inhomogeneous bilayer that buckles out of plane to reach a 3D static shape when the fabric is released from its frame (Figure 1). We examine the interplay of geometry and physics that governs the behavior of the ribbon-fabric ensemble, and use those insights to develop an algorithm for inverse design and a fabrication pipeline for approximating a given freeform surface by a self-actuating deployable structure.

Relation to Previous Plastic-on-Fabric Techniques Several recent methods have experimented with the idea of printing plastic on fabric to rapidly prototype lightweight curved surfaces [GC16; POT17]. The main difference of our approach is that we propose to print *dense, broken ribbons* of plastic, rather than the *sparse networks of closed curves* found in prior work.



While both strategies involve printing plastic on fabric, there are deep differences in the mechanics of how the plastic-fabric ensemble buckles, and in the ability of the ribbons to control the shell geometry. Specifically, in a sparse assembly of long ribbons (as found in prior work), most of the shape change during self-actuation is the result of compression of the pre-stretched fabric in the regions *in between* the curves. The curves buckle and bend to release this stress [POT17] (inset, top), and serve as boundary conditions for the fabric which shrinks to a Plateau-like surface. On the one hand, this mechanism allows for significant coarse-scale shape change with only a small amount of added plastic material. On the other hand, the sparse ribbons provide only limited control over the geometry of the fabric patches (which always have negative intrinsic and approximately zero extrinsic curvature).

In contrast, in our dense assemblies, the plastic ribbons cover a significant fraction of the fabric surface area: not only does the shrinking fabric exert in-plane stress on the plastic ribbons, but *bilayer effects* become important in the regions covered by plastic. As the fabric relaxes and shrinks to relieve stretch, it exerts stress on the plastic along the plastic-fabric interface. Since the plastic is incompressible yet elastic, the ribbon buckles to form an arc at the equilibrium state (inset, bottom). The curvature of this arc depends on the pre-stress of the fabric as well as on the thicknesses of the elastic and plastic layers. At the coarse scale, our dense assemblies thus form a type of metamaterial, whose geometry is directly controlled by the spacing, orientation, and thickness of the plastic ribbons. Given a sheet of fabric with constant pre-stress, our key idea is to *program* the curvature of a target surface by optimizing these parameters to leverage the above mechanisms of shape change.

Note that in contrast to methods that mainly rely on metric frustration to control *intrinsic* curvature [KPCP18; CPSP21], our approach provides control of *extrinsic* curvature as well, as we can force the surface to bend along prescribed directions by aligning ribbons with those directions. The ability to manipulate extrinsic curvature is particularly important when fabricating nearly-developable surfaces, when prescribing Gaussian curvature is not enough to reliably achieve the desired shape.

Overview Our inverse design tool takes as input a triangle mesh embedded in $\mathbb{R}^3$ representing the desired target (deployed) surface shape, and computes (1) a flattening of this surface into the plane, and (2) a set of ribbons over this planar domain, so that 3D-printing the ribbons onto fabric with constant pre-stress, and cutting the fabric along the boundary of the planar domain, yields an assembly whose static shape deploys to match the target surface (see Figure 2). The task of programming the desired surface curvature is divided between these two steps: during flattening, we optimize for a ribbon layout that will achieve the desired *intrinsic* curvature, and aligns with directions of principal *extrinsic* curvature to prevent torsion of the ribbons. Then, we choose ribbon thicknesses to induce the desired extrinsic curvature magnitude, based on a data table constructed by measuring physical samples.

We compute the flattening by optimizing for a 2D parameterization of the target surface, subject to certain fabricability constraints:

1. the metric distortion of the parameterization remains below the maximum stretch of the fabric;
2. the metric distortion of the parameterization is close to 1 along the direction of maximum curvature.

The first condition encodes that the fabric can, at most, remain in its pre-stretched state, and cannot stretch further. The second condition ensures that the assembly will undergo little compression along directions of high curvature, which allows us to print closely-spaced, incompressible plastic ribbons along those directions to maximally exploit the bilayer effect to induce the necessary surface curvature. We solve for a flattening that satisfies these properties using a local/global iterative algorithm akin to As-Rigid-As-Possible parameterization [LZX*08]. While recent methods relied on a similar procedure to bound distortion for other fabrication techniques [AXZ*18; PIC*21], we introduce a hard constraint on the *direction* of distortion to align our plastic ribbons with the principal directions of curvature of the target surface.

We then position the plastic ribbons over the flattened surface to form a dense pattern with spatially-varying spacing and thickness. As explained above, the resulting pattern acts as a homogenized metamaterial, where spacing between the ribbons controls the average stretch of the assembly as dictated by the parameterization, while thickness controls the magnitude of normal curvature. We generate G-code for printing this ribbon pattern by tracing orthogonal families of curves along smooth curvature-aligned direction fields [KCPS13; KCPS15].

Since we use a 3D printer to deposit plastic on top of the stretched fabric, the bilayer effect strongly biases the surface to curve away from the printer bed, so that the curvature of the assembly after it has relaxed to its static shape is always positive in the direction of the ribbons. We circumvent this limitation by introducing a custom, reversible frame structure that allows us to print

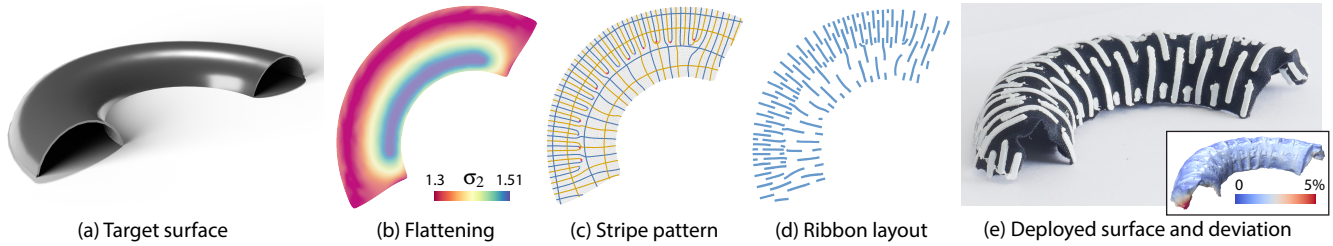


Figure 2: *Overview of our method. Given a target surface (a), we first compute its principal curvatures, along which plastic ribbons will be placed. We compute a flattening of the 3D surface into the plane, with bounded stretching (b; colors indicate the magnitude of stretching σ_2 in the direction perpendicular to the ribbons). We then generate a staggered stripe pattern aligned with these directions; the deformation desired for deploying from the flattened state to 3D determines the spacing and density of the ribbon pattern (c). We generate plastic ribbons from this pattern, and adjust their thicknesses according to the target extrinsic curvature magnitude (d). Printing these plastic ribbons on pre-stretched fabric results in a self-actuated assembly that deploys to a shape approximating the target surface when released (e).*

plastic on both sides of the stretched fabric. This simple hardware further expands the range of surfaces we can reproduce (Figure 9).

Contributions In summary, we introduce:

- a new mechanism for self-actuated structures, based on closely-spaced plastic ribbons printed on pre-stretched fabric, which provides controllable directional distortion and curvature;
- an inverse design tool to fabricate low-cost, lightweight prototypes of freeform surfaces using this mechanism;
- a parameterization algorithm that bounds surface distortions along and across prescribed directions, along with a pattern synthesis algorithm that covers a surface with ribbons to match the target distortions and curvature given by the aforementioned parameterization;
- a simple hardware setup to reliably print plastic ribbons on both sides of a pre-stretched sheet of elastic fabric, with minimum manual intervention.

We applied our approach to fabricate a number of freeform surfaces, and evaluate our method by measuring the agreement between the fabricated and target 3D shapes.

2. Related Work

The need to manufacture 3D surfaces out of flat sheets of material has motivated significant research in computer graphics, architectural geometry, and materials science. Representative examples include work on materials that can bend but cannot shear or stretch—like paper—to model *developable surfaces* [TBWP16; KMM17; RHS17; SGC18; SC18; IRHS20]; materials that have the additional ability to shear, such as *Chebyshev nets* [GSD*14] and other forms of *gridshells* [PKI*19; dPel17; PLBM20]; and assemblies that can locally expand or shrink to induce intrinsic curvature [KCD*16; MPI*18; CPSP21].

Of particular interest for us are materials that can lift to 3D without external forces, resulting in so-called *self-actuated* assemblies. Such properties are typically obtained thanks to materials that produce internal stress under changes of temperature [BvRL*19; NBB*16; KHB*12; GMP*20] or moisture [GMN*16], or materials that have been pre-stressed as part of the flat manufacturing process, such as elastic sheets of latex or fabric [GMB17; POT17]. As explained in the introduction, we follow the latter approach, and rely on competition between residual stress in pre-stretched fabric and inextensibility of plastic ribbons to drive shape change. This mechanism relies on commodity hardware (a desktop 3D printer and fabric), which contrasts with alternatives that require special lab equipment to glue plastic tiles over a pre-stretched latex sheet [GMB17]. The geometry of surfaces that buckle in response to anisotropic material changes due to these mechanisms has been studied extensively in the physics community [SE10; PSJH17; GAE19], though much of this analysis assumes a single anisotropic material, rather than a patterned composite of two materials.

Another recently-proposed mechanism for self-actuation is inflation of air channels [PIC*21]: two thin, flat sheets of elastic material are fused along a network of curves, creating air pockets between the layers in the form of tube-like channels. Pressurizing the interstitial space causes the channels to inflate and contract transversally. At the highest level, the design problem for inflatables solved by Panetta et al. shares similar features to the problem we solve for ribbon networks on fabric: in both cases, actuation results from programming anisotropic residual strain in a flat sheet, parameterized by a network of curves, and subject to fabricability constraints on the maximum strain. But crucial differences prevent using the method of Panetta et al. for designing ribbon layouts: inflated channels have zero normal curvature, whereas ribbons exhibit strong extrinsic curvature (due to the bilayer effect) which can be controlled and must be accounted for; moreover, whereas inflated channels have circular cross-section, plastic ribbon cross-sections are rectangular and a ribbon layout must account for potential axial twisting of the ribbons due to geodesic torsion (see Section 3 and Figure 3).

Inspired by early experiments in design and architecture [OR07; GC16; BAB20; Kyc19; Fie18], several computational design systems have been proposed to simulate assemblies of plastic curves embedded in elastic membranes [POT17; JSVB21]. Pérez et al. [POT17] consider objects composed of a sparse network of curves, resulting in so-called *Kirchhoff-Plateau surfaces* where the membrane behaves like a minimal surface in-between the curves. However, sparse networks are difficult to design automatically, which is why Pérez et al. adopt a direct modeling workflow, offering only

a local inverse design tool to modify existing layouts. Moreover, since their assemblies are dominated by empty fabric, Pérez et al. ignore the forces exerted by the membrane along the curves, where the plastic and fabric form a *bilayer* structure (see also the introduction for more discussion of this line of work). Jourdan et al. [JSVB21] propose a physical model to simulate this structure, allowing them to better reproduce the arched shape taken by individual plastic segments in reaction to the fabric forces underneath. However, they use their model for the forward simulation of specific assemblies of star-shaped patterns, while we target an inverse design task where we want to automatically find a pattern of plastic ribbons that would reproduce a target shape when fabricated.

Surface parameterization is a classic problem in geometry processing [HLS07], with many available algorithms that seek to minimize angle [SLMB05; SC17] or length distortion [ZSGS04], while possibly also considering ancillary complications such as seam placement, local and global injectivity, atlas decomposition, etc. We do not give a complete survey here; note that while conformal mapping in particular is popular for texture mapping and similar applications, and for fabrication techniques that exploit isotropic scaling [KCD*16; GMB17], we specifically require an *anisotropic* parameterization with bounded distortion. Few prior methods approached the problem of computing a flattening whose distortion is bounded in one *arbitrary* direction [AXZ*18]. Closest to our approach is Panetta et al. [PIC*21]'s flattening step, which includes a soft penalty term to favor alignment of the channels perpendicularly to the principal direction of curvature. In contrast, we treat curvature direction alignment as a hard constraint, which allows us to build our flattening algorithm on top of As-Rigid-As-Possible (ARAP) parameterization [LZX*08], a classic and easily-extendable local-global algorithm for distortion-minimizing parameterization. See Section 4.1 for more discussion of the pros and cons of the two approaches.

The last step of our method consists in generating a pattern of plastic ribbons over the flattened shape, where the local orientation and spacing of the ribbons is dictated by the curvature and distortion fields of the target surface. Producing such a graded pattern of discrete, elongated elements arises in other fabrication scenarios, including the design of metal frame and wire structures [MLB16; MWS*20], knitting paths [NAH*18], microstructures [TTZ*20], weaves [VZF*19; RPC*21] and nets [SCBV19], fiber-reinforced composites [BTM*20], and the aforementioned inflatable structures [PIC*21]. Many of the algorithms in geometry processing on vector-field integration could be used to generate ribbon curves from our curvature-aligned vector field; see the survey by Vaxman et al. [VCD*16] for a review of some of these alternatives. We chose the method of Knoppel et al. [KCPS15], originally designed for computing stripe textures on surfaces, due to the ease of specifying the frequency of the reconstructed pattern in the transverse direction, and availability of source code.

3. The mechanics of plastic ribbons embedded in stretched fabric

The key idea behind our approach is to balance between two effects that occur when plastic ribbons are bound to a stretched fabric substrate. First, the ribbons frustrate the contraction of the fabric back to its rest state, forcing residual strain in the fabric and creating intrinsic curvature. Second, as mentioned in the introduction, due to differential compression of the plastic and fabric layers in the direction of the ribbons, there is a *bilayer effect* that induces each ribbon to buckle into an arc when the assembly is released. We seek to exploit both phenomena to program the desired surface curvature.

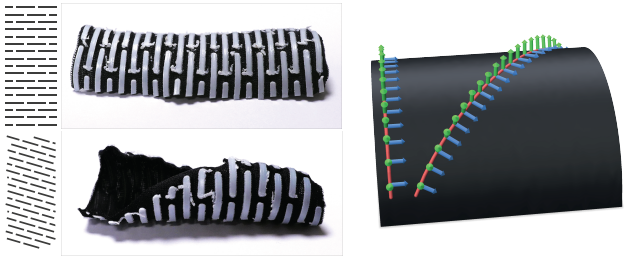
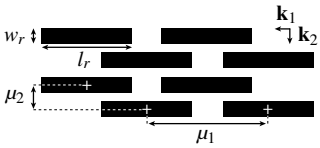


Figure 3: *Due to the bilayer effect, parallel plastic ribbons roll to form a cylinder (a, top). Attempting to orient the ribbons away from the direction of maximum curvature still yields a cylinder as ribbons resist torsion (a, bottom). On a curved surface, curvature lines are the only curves with zero geodesic torsion, as visualized by sliding a Darboux frame along the curve and observing its rotation around the curve tangent (b, after [IBB15]).*

To that end, we define a pattern of plastic ribbons that behaves like a homogeneous metamaterial with controllable stretch and curvature, illustrated as inset. This pattern will be printed on fabric that has been uniformly stretched by a factor s. In this pattern, the plastic ribbons are separated by empty fabric to form dashed stripes. Parallel stripes are also separated by empty fabric. Note that we shift every two stripes by half the period of the dashes to form a staggered pattern, which prevents the emergence of long bands of empty fabric transverse to the plastic ribbons. Varying the quantity of empty fabric in-between consecutive and parallel plastic ribbons controls the average contraction $1/\sigma_1$ and $1/\sigma_2$ of the metamaterial when deployed, while varying the thickness τ of plastic deposited on the ribbons impacts the strength of their bilayer effect.



Given a target surface $\mathcal{S}$, our key idea is to design an appropriate ribbon pattern in three stages:

1. The bilayer effect induces significant extrinsic curvature in the direction of the ribbons, and the plastic ribbons resist torsion along that direction (Fig. 3a). Based on these two observations, we align the ribbon pattern (horizontal axis in the inset figure) in the direction of maximum magnitude principal curvature $\mathbf{k}_1$, which maximally exploits the bilayer effect and minimizes torsion along the ribbons since curvature lines are characterized by vanishing geodesic torsion [dCar76; BFS10; IBB15] (Fig. 3b). It follows that the transverse direction of the pattern aligns to the other principal curvature direction $\mathbf{k}_2$.
2. When the plastic-fabric ensemble is released and allowed to relax to equilibrium, the fabric will contract. At most, it will return

to its original shape; and at least, it will not contract at all (if the fabric is covered completely in plastic). The fabric pre-stretch factor s, together with fabricability constraints on the minimum and maximum values of ribbon length and width l_r, w_r and spacing μ_1, μ_2, determine the range of possible contractions in the ribbon and transverse direction. We compute a flattening of the target surface $\mathcal{S}$ to the 2D plane, which satisfies these constraints (Section 4).

3. To control extrinsic bending of the surface, we adjust the ribbon thickness τ, using a data-driven law for the relationship between τ and curvature derived from physical measurements (Section 5). In practice, we only measured patterns of parallel ribbons, which makes the resulting law better suited to surfaces that are nearly developable rather than doubly-curved.

4. Controlling stretching

We first describe our flattening formulation in a continuous setting, and then describe its discretization.

4.1. Flattening with bounded directional scaling

Let the mapping $\varphi\colon \Omega \subset \mathbb{R}^2 \to \mathbb{R}^3$ describe our (given) target deployed surface $\mathcal{S}$, and let $\bar{\varphi}\colon \Omega \subset \mathbb{R}^2 \to \mathbb{R}^2$ describe its (currently unknown) flattened, pre-stretched counterpart $\bar{\mathcal{S}}$, with Ω denoting an arbitrary surface parametric domain with coordinates (u,v). We write $\mathbf{k}_1(u,v), \mathbf{k}_2(u,v)$ for the vector fields $\Omega \to \mathbb{R}^3$ in the direction of maximum-magnitude and minimum-magnitude principal curvature of $\mathcal{S}$.

We seek a flattening $\bar{\varphi} \circ \varphi^{-1} : \mathcal{S} \to \bar{\mathcal{S}}$ with the following properties: (1) the principal stretch directions are aligned with $\mathbf{k}_1, \mathbf{k}_2$; (2) the two principal stretches σ_i are in the range $1 \leq \sigma_i^{\min} \leq \sigma_i \leq \sigma_i^{\max}$. As discussed above, the achievable range of principal stretches $[\sigma_i^{\min}, \sigma_i^{\max}]$ is determined by s, the geometry of the ribbon pattern, and limits on the accuracy and resolution of the 3D printer; we discuss how to determine these ranges in Section 4.3, and assume they are given for now (roughly speaking, σ_1 must be close to inextensible, while there is more flexibility in the choice of σ_2, since the ribbons are longer than they are wide).

More precisely, let $J_{\bar{\varphi}} = \mathrm{d}\bar{\varphi} \in \mathbb{R}^{2\times 2}$ denote the Jacobian of the mapping $\bar{\varphi}$, and likewise for J_φ. We are looking for a mapping $\bar{\varphi}$ such that the pushforward $\mathrm{d}(\bar{\varphi} \circ \varphi^{-1})$ has the form

$$\mathrm{d}(\bar{\varphi} \circ \varphi^{-1}) = R_{\bar{\mathcal{S}}} \Sigma R_{\mathcal{S}}^t, \forall (u,v) \in \Omega, \tag{1}$$

where $R_{\bar{\mathcal{S}}}(u,v)$ is an arbitrary rotation matrix; $R_{\mathcal{S}} = \begin{bmatrix}\hat{\mathbf{k}}_1 & \hat{\mathbf{k}}_2\end{bmatrix}_{3\times 2}$ rotates the Euclidean plane to the tangent plane of $\mathcal{S}$, with the Euclidean axes mapping to the principal curvature directions; and $\Sigma(u,v) = \begin{pmatrix}\sigma_1(u,v) & 0 \\ 0 & \sigma_2(u,v)\end{pmatrix}$ encodes stretching, subject to the bound constraints $\sigma_i \in [\sigma_i^{\min}, \sigma_i^{\max}]$.

Since satisfying (1) exactly for all $(u,v) \in \Omega$ might be impossible, we seek the closest solution in the least square sense. Denoting $\|\cdot\|_F$ the Frobenius norm, we cast our minimization problem as

$$\min_{\bar{\varphi}, \sigma_1, \sigma_2, R_{\bar{\mathcal{S}}}} \int_\Omega \|J_{\bar{\varphi}} - R_{\bar{\mathcal{S}}} \Sigma R_{\mathcal{S}}^t J_\varphi\|_F^2 \, \mathrm{d}u\mathrm{d}v, \tag{2}$$

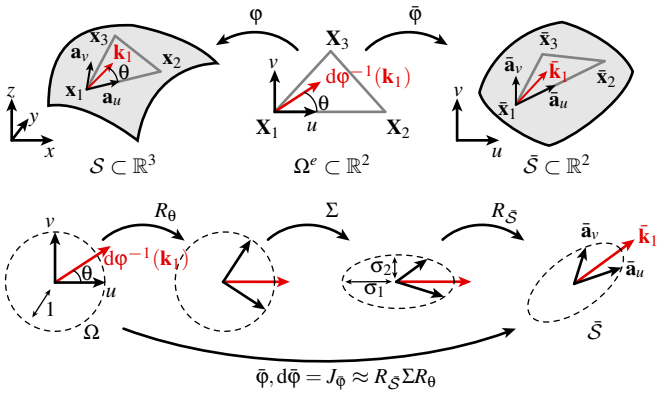


Figure 4: *Overview of the flattening algorithm. Each triangle of the target mesh $\mathcal{S}$ is first mapped to the 2D plane by aligning its $(\mathbf{x}_1, \mathbf{x}_2)$ edge with the u-axis of our 2D Cartesian coordinate system. We then compute its mapping $\bar{\varphi}$ to the flattened configuration $\bar{\mathcal{S}}$ using a local/global algorithm (top). This mapping $\bar{\varphi}$ is built so as to maximally stretch the surface $\mathcal{S}$ along its direction of maximum curvature $\mathbf{k}_1$ (bottom).*

whose unknowns are the fields $\bar{\varphi} : \Omega \to \mathbb{R}^2$, $\sigma_1 : \Omega \to [\sigma_1^{\min}, \sigma_1^{\max}]$, $\sigma_2 : \Omega \to [\sigma_2^{\min}, \sigma_2^{\max}]$ and $R_{\bar{\mathcal{S}}} : \Omega \to SO(2)$.

While this formulation is reminiscent of other flattening methods with bounded scaling [AXZ*18; PIC*21], unique to our approach is the use of the fixed matrix $R_{\mathcal{S}}$ to strictly constrain the directions of maximal strain. In contrast, Panetta et al. [PIC*21] employ a *soft* regularizer to penalize alignment of their air channels with directions of high normal curvature. While their formulation gives additional freedom to trade curvature alignment for better scaling distribution, the strong resistance of plastic ribbons to torsion (Fig. 3) demands close curvature alignment and limits the usefulness of this tradeoff. Moreover, hard constraints allow the use of ARAP-style local-global iterations (described next), which are simpler and easier to implement than the black-box nonlinear optimization using Newton's method that would be required for soft constraints.

4.2. Discrete formulation

We discretize (2) using triangular elements. To this end, we represent the surfaces $\mathcal{S}$ and $\bar{\mathcal{S}}$ using triangle meshes having same number of vertices $n_{\mathcal{V}}$ and faces $n_{\mathcal{T}}$, and same topology, and we stack the coordinates of their vertices into the vectors $\mathbf{x}$ and $\bar{\mathbf{x}}$ respectively. Approximating $\bar{\varphi}$ by a piecewise linear function makes $J_{\bar{\varphi}}$ constant on each triangle. We denote by J^e the value of $J_{\bar{\varphi}}$ on element e, and likewise for other quantities such as $\Sigma^e = \begin{pmatrix}\sigma_1^e & 0 \\ 0 & \sigma_2^e\end{pmatrix}$, etc.

Note that in practice we do not need to build a full (and consistent) parametrization of the surface $\mathcal{S}$ on the entire domain Ω at once to compute the Jacobians J^e; we can instead locally parametrize the surface on a per triangle basis. More specifically, we define the preimage Ω^e of a given element e of $\mathcal{S}$ with 3D vertex coordinates $\mathbf{x}_{1\leq k\leq 3}^e$ as a 2D triangle of same shape and size with vertex coordinates $\mathbf{X}_k^e$, whose edge vector $\mathbf{X}_2^e - \mathbf{X}_1^e$ is aligned with the u-axis of our 2D Cartesian coordinate system (see Figure 4). We

can then express the Jacobian J^e as $J^e = [\bar{\mathbf{x}}_2^e - \bar{\mathbf{x}}_1^e \quad \bar{\mathbf{x}}_3 - \bar{\mathbf{x}}_1^e][\mathbf{X}_2^e - \mathbf{X}_1^e \quad \mathbf{X}_3^e - \mathbf{X}_1^e]^{-1}$. Moreover, on each triangle, the parameterization of $\mathcal{S}$ is now an isometry, with $R_{\mathcal{S}}^t J_\varphi = R_\theta$ a 2×2 rotation matrix by angle θ, where θ is the angle from $\mathrm{d}\phi^{-1}\mathbf{k}_1$ to the u-axis.

We can now rewrite our optimization problem (2) as

$$\min_{\bar{\mathbf{x}},\sigma_1,\sigma_2,R} \underbrace{\sum_{i=1}^{n_\mathcal{T}} \|J^i(\bar{\mathbf{x}}) - R^i_{\bar{\mathcal{S}}}\Sigma^i R^i_\theta\|_F^2 \, A_i}_{E(\bar{\mathbf{x}},\sigma_1,\sigma_2,R)}, \tag{3}$$

where $\sigma_1 = \{\sigma_1^i \in [\sigma_1^{\min},\sigma_1^{\max}]\}$, $\sigma_2 = \{\sigma_2^i \in [\sigma_2^{\min},\sigma_2^{\max}]\}$, $R = \{R^i_{\bar{\mathcal{S}}} \in SO(2)\}$, and A_e is the area of element e in the mesh corresponding to $\mathcal{S}$.

To solve problem (3), we adapt the local/global optimization algorithm proposed by Liu et al. [LZX*08] and alternate between local steps in which we optimize the per-triangle rotations and stretches $R^e_{\bar{\mathcal{S}}}$ and Σ^e, and global steps where we minimize the cost function E over the positions $\bar{\mathbf{x}}$ while keeping all the matrices $R^e_{\bar{\mathcal{S}}}$ and Σ^e fixed. We detail below how we solve these local and global problems.

Computation of stretching matrices Σ^e . For each element e, the minimizer of E with respect to σ_1^e and σ_2^e can be obtained by solving the local problem

$$(\tilde{\sigma}_1^e,\tilde{\sigma}_2^e) = \operatorname*{argmin}_{\substack{(\sigma_1^e,\sigma_2^e)\\ \sigma_1^{\min}\le\sigma_1^e\le\sigma_1^{\max},\\ \sigma_2^{\min}\le\sigma_2^e\le\sigma_2^{\max}}} \|J^e(\bar{\mathbf{x}}) - R^e_{\bar{\mathcal{S}}}\Sigma^e R^e_\theta\|_F^2. \tag{4}$$

As shown in Appendix A, we can derive closed-form expressions for $\tilde{\sigma}_1^e$ and $\tilde{\sigma}_2^e$:

$$\tilde{\sigma}_i^e = \begin{cases} \sigma_i^{\min} & \text{if } \hat{\sigma}_i^e < \sigma_i^{\min}, \\ \hat{\sigma}_i^e & \text{if } \sigma_i^{\min} \le \hat{\sigma}_i^e \le \sigma_i^{\max}, \\ \sigma_i^{\max} & \text{if } \sigma_i^{\max} < \hat{\sigma}_i^e, \end{cases} \quad i = \{1,2\}, \tag{5}$$

where $\hat{\sigma}_i^e = [R^e_\theta (J^e)^t R^e_{\bar{\mathcal{S}}}]_{ii}$.

Computation of rotation matrices $R^e_{\bar{\mathcal{S}}}$. Minimizing E with respect to $R^e_{\bar{\mathcal{S}}}$ amounts to solving the local problem

$$\tilde{R}^e_{\bar{\mathcal{S}}} = \operatorname*{argmin}_{R^e_{\bar{\mathcal{S}}}\in SO(2)} \left\|J^e(\bar{\mathbf{x}}) - R^e_{\bar{\mathcal{S}}}\Sigma^e R^e_\theta\right\|_F^2. \tag{6}$$

We show in Appendix B that the optimal rotation matrix $\tilde{R}^e_{\bar{\mathcal{S}}}$ is given by

$$\tilde{R}^e_{\bar{\mathcal{S}}} = UV^t, \tag{7}$$

where U and V are the orthogonal matrices of the singular value decomposition of the matrix $J^e(R^e_\theta)^t\Sigma^e$, up to flipping the sign of the column of U corresponding to the smallest singular value so that $\det(UV^t) > 0$.

Computation of 2D node positions $\bar{\mathbf{x}}$. Following Liu et al. [LZX*08] and denoting $M^e = R^e_{\bar{\mathcal{S}}}\Sigma^e R^e_\theta$, we rewrite our energy E using cotangent weights as

$$E = \sum_{(i,j)\in\mathcal{H}e} \frac{\cot\alpha_{ij}}{2} \left\|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j - M^{\mathcal{T}(i,j)}(\mathbf{X}_i - \mathbf{X}_j)\right\|^2, \tag{8}$$

where $\mathcal{H}e$ denotes the set of all half-edges of the meshes, $\mathcal{T}(i,j)$ the triangle incident to the half-edge (i,j), and α_{ij} the angle opposite to the half-edge (i,j) in the triangle $\mathcal{T}(i,j)$.

The minimum of (8) with respect to $\bar{\mathbf{x}}$ can then be obtained by solving the linear system

$$\begin{aligned} &\sum_{j\in\mathcal{N}(i)} \left[\cot\alpha_{ij} + \cot\alpha_{ji}\right](\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j) \\ &= \sum_{j\in\mathcal{N}(i)} \left[\cot\alpha_{ij} M^{\mathcal{T}(i,j)} + \cot\alpha_{ji} M^{\mathcal{T}(j,i)}\right](\mathbf{X}_i - \mathbf{X}_j) \end{aligned} \tag{9}$$

for all vertices i, where $\mathcal{N}(i)$ represents the set of vertices adjacent to i.

4.3. Implementation details

Computation of curvature directions $\mathbf{k_1}$. We compute the directions of maximum curvature $\mathbf{k_1}$ on the target mesh $\mathcal{S}$ using the method of Knoppel et al. [KCPS13] that we slightly modified to output a line field aligned with the curvature directions having largest magnitude in terms of *absolute value*. For many examples, we kept the default value $\lambda = 0$ for the parameter controlling the tradeoff between faithfulness to the curvature directions and smoothness of the output field (smoothness is necessary to regularize the field when the principal curvature directions have similar magnitudes). However, in some cases, we found necessary to slightly decrease this value to avoid oversmoothing (see Table 1).

Setting the principal stretch bounds $\sigma^{\min},\sigma^{\max}$. The upper bound in the transverse direction is determined by how much spacing we allow between neighboring ribbons. We fix this distance to $\mu_2^{\max} = l_r$ to avoid producing large areas free of plastic. From this value, we deduce $\sigma_2^{\max} \approx 1.51$ using Equation 10 described in Section 6.1. For the lower bound, we set the value $\sigma_2^{\min} = 1.3$ as a safeguard against fusion of neighboring ribbons. Along $\mathbf{k}_1$, we set $\sigma_1^{\min} = 1$ and $\sigma_1^{\max} = 1.3$ to give room to the fabric to contract a bit along the principal direction of curvature if necessary to flatten the surface.

Initialization of $\bar{\mathbf{x}}$, σ_1, σ_2 and R. We initialize the vertex positions $\bar{\mathbf{x}}$ of the flattened mesh by computing a Tutte's embedding after regularly distributing the boundary vertices on a unit disk. We then run the implementation of *Scalable locally Injective Mappings* [RPPS17] (as implemented within libigl [JP*18]). The initial stretch values σ_1^e, σ_2^e are set to 1. The rotations matrices $R^e_{\bar{\mathcal{S}}}$ are initialized using Formula (7).

Update scheme and termination. We alternate between local computations of R, σ_1 and σ_2 and global computation of $\bar{\mathbf{x}}$ (in this order). The loop updating R, σ_1 and σ_2 is run twice before switching to $\bar{\mathbf{x}}$. The algorithm is stopped when the average change of both σ_1 and σ_2 is below 10^{-6}. Note that the principal stretches and stretch directions that will be used subsequently to compute ribbon layout are those of the Jacobians J^e at termination, which might not exactly satisfy the scaling bounds if the optimal residual of Equation 3 is nonzero. See Table 1 for detailed statistics about the behavior of the optimization in practice.

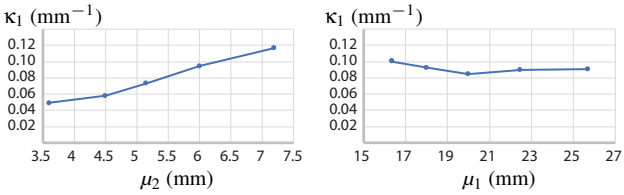


Figure 5: *The curvature κ_1 of our metamaterial varies not only as a function of plastic thickness τ, but also as a function of spacing μ_2 in the transverse direction (left, with $\mu_1 = 22.5$mm and $\tau = 0.5$mm). In contrast, varying spacing μ_1 in the ribbon direction has little impact (right, with $\mu_2 = 6$mm).*

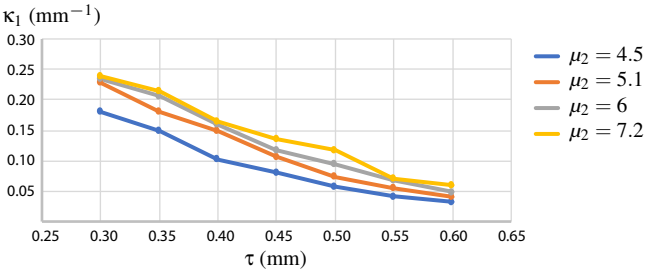


Figure 6: *Curvature κ_1 of our metamaterial as a function of plastic thickness τ and spacing μ_2 (in mm).*

5. Controlling Curvature

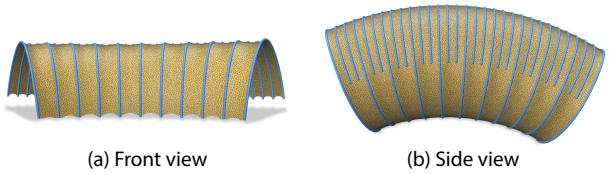
As discussed in the previous section, we align the ribbon pattern along the direction of maximum principal curvature $\mathbf{k}_1$ to take advantage of the bilayer effect: the tendency of plastic printed on top of pre-stressed fabric to bend about the axis transverse to the ribbon, to relieve the differential stress in the thickness direction at the bilayer interface. Jourdan et al. [JSVB21] derived an expression for the curvature expected for an *isolated* plastic ribbon printed on fabric, as a function of the ribbon thickness. However, when we tested the theory using physical prototypes, we observed that curvature is also affected by the quantity of empty fabric surrounding the ribbons, as this fabric exerts additional forces on the ribbons and modifies their equilibrium shape.

We conducted an experiment where we printed several regular patterns that roll in a cylindrical shape (inset), with ribbons of constant thickness, width and length, but varying μ_1 and μ_2. We then measured the curvature κ_1 of each cylinder, as plotted in Figure 5. This experiment reveals that curvature is affected primarily by the amount of empty fabric in the *transverse* direction, μ_2, and not by the spacing in the ribbon direction, μ_1. We conclude from this experiment that the forces applied by the fabric along the ribbons depend on the area of fabric *across* the ribbon (controlled by μ_2).

Given the complex interplay between the fabric and the ribbons forming our assemblies, we chose to adopt a data-driven approach to relate the target curvature to the thickness and spacing of the ribbons. Concretely, we printed a series of regular patterns with varying plastic thickness τ and spacing μ_2, while holding μ_1 and the ribbon width and length fixed. We then measured the curvature κ_1 along the $\mathbf{k}_1$ direction of the resulting cylinders, as reported in Figure 6. Since the function is monotonic, tabulating and interpolating this data allows us to obtain, for a target curvature and spacing, the thickness τ that should be used for each ribbon. We only measured this data on cylindrical shapes produced by parallel patterns, as for most surfaces, the method of Knoppel et al. [KCPS13] produces curvature fields that are locally near-parallel, but our results demonstrate that our approach for assigning thickness generalizes to more complex shapes too (including shapes in which ribbon orientation changes slowly over a large area (Fig. 1, Fig. 2) and shapes where there are orientation singularities, but localized to small regions (Fig. 9, first row)). While additional data could be collected for doubly-curved samples, this would come at the cost of a more complex measurement procedure, and a higher-dimensional data table.

We also considered physical simulation to optimize for the ribbon parameters that reproduce the target shape. However, our experiments with existing models [POT17; JSVB21] revealed that they cannot reliably predict the coarse-scale behavior of our ribbon patterns. Because the plastic ribbons bond to the fabric along their entire width, the presence of ribbons causes the surface to resist transverse contraction, by a factor that depends on the ribbon width w_r (See Eq. 10 below). Rod-shell coupling models that assume that the ribbons are infinitesimally wide fail to capture this phenomenon. Fig. 7 shows the result of the model by Jourdan et al. [JSVB21] when simulating a similar pattern as in Fig. 2. The simulation predicts that the shell contracts uniformly during deployment, yielding an equilibrium shape different from the target half-torus. (In reality, the shell contracts more along the interior boundary than along the exterior one.) Other effects that are not captured by existing simulations, that we hypothesize are important to accurately predicting the deployed shape, include nonlinearity of the fabric (which is knitted, and stretched well outside its linear regime); plasticity of the ribbons; residual stress in the ribbons induced by the printing process; printer fabrication errors; and seeping of the plastic partially into the fabric during printing. While more complex models might capture such effects, their development would require extensive lab testing and parameter sweeps to be repeated for any change in the fabric/printer/filament used. In contrast, our approach based on geometric principles and data fitting allows good control over the deployed shape while being fast and relatively simple.

Front-and-back asymmetry The fabric we used in our experiments was knitted with a *stockinette* stitch pattern, which has a stronger bending resistance in one direction than in the other. We found that the curvature varied significantly for similar values of τ and μ_2 depending on the side of the fabric we printed on: the same ribbons, when printed on the stitch front, had a 50 to 60% higher curvature than the ones printed on the stitch back. We therefore computed two different tables of μ_2 vs. κ_1 vs. τ, one for each side of the fabric, each of these tables has 8×6 data points in total.



(a) Front view (b) Side view

Figure 7: *The rod-based model of Jourdan et al. [JSVB21] does not account for the width of the plastic ribbons. When attempting to simulate a similar pattern as in Fig. 2, the surface bends properly along the ribbons due to the bilayer effect (a), but it contracts uniformly in the direction transverse to the ribbons, despite the varying density of plastic along the interior and exterior boundary (b). As a result, the simulation predicts a deployed shape that does not match the target half-torus.*

6. Ribbon layout

We are now equipped with a flattened surface $\bar{\mathcal{S}}$, along with the directions $\bar{\mathbf{k}}_i = \mathrm{d}\left(\bar{\varphi}\circ\varphi^{-1}\right)\mathbf{k}_i$ and magnitudes σ_i of principal stretch of the flattening map. Our goal is to map these quantities to the parameters l_r, w_r, μ_1, and μ_2 of the discrete ribbon pattern that will be printed on the fabric (see inset figure in Section 3), and compute curves on $\bar{\mathcal{S}}$ that trace the centerline of each ribbon. Together with the thickness τ computed in Section 5, these parameters and curves fully determine the structure to be fabricated.

6.1. Determining the Pattern Parameters

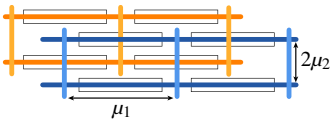
In theory, we can vary the quantity of empty fabric locally by adjusting the length l_r and width w_r of the ribbons, or the spacing μ_1 and μ_2 between the centers of neighboring ribbons, or both. In other words, a dense pattern of thin ribbons yields as much empty fabric as a sparse pattern of large ribbons. Since varying the spacing between ribbons is less susceptible to limitations on printer accuracy and resolution than attempting to vary the ribbon dimensions, we eliminate this redundancy by fixing l_r and w_r, leaving spacing between the ribbons along μ_1 and μ_2 as the only parameters that control stretch. We experimentally set $l_r = 15\text{mm}$ and $w_r = 1.5\text{mm}$ as a trade-off between the resolution of the pattern and the adherence of the ribbons, as smaller ribbons would increase resolution but adhere less to fabric.

Given the fabric pre-stretch s, we compute the values of spacing μ_1 and μ_2 to achieve the target stretching values σ_1 and σ_2 by considering individual stripes of ribbon dashes. More specifically, we assume that, after we release the fabric, a periodic motif of initial length μ_1 contracts to an average length $\bar{\mu}_1 = l_r + \frac{\mu_1 - l_r}{s}$. Likewise, we estimate the average width of parallel motifs of initial width μ_2 to $\bar{\mu}_2 = w_r + \frac{\mu_2 - w_r}{s}$. Taking the ratios of initial to contracted lengths (respectively widths) gives us the average stretch values $\sigma_1 = \frac{\mu_1}{\bar{\mu}_1}$ and $\sigma_2 = \frac{\mu_2}{\bar{\mu}_2}$, from which we deduce

$$\mu_1 = l_r \frac{s-1}{s-\sigma_1}\sigma_1, \qquad \mu_2 = w_r \frac{s-1}{s-\sigma_2}\sigma_2. \tag{10}$$

Note that these equations assume that the empty fabric surrounding the ribbons effectively contracts by a factor of s on deployment. In practice, the presence of staggered ribbons on each side of the gaps along $\bar{\mathbf{k}}_1$ might prevent Equation (10) from holding for small values of μ_2. Nevertheless, we found this effect to be negligible once the bound $\sigma_2^{\min}$ on σ_2 was enforced during optimization.

6.2. Generating the ribbon layout



As discussed above, we reduced the design space of our ribbon pattern to three parameters that correspond to the spacing μ_1 and μ_2 of the ribbons, and their thickness τ. We now need to place ribbons over the flattened surface $\bar{\mathcal{S}}$ so that their local spacing agrees with the target stretching values σ_1 and σ_2 according to Equation (10). We achieve this goal by leveraging the fact that the ribbons in our staggered pattern lie on two grid layouts with cell size $\mu_1 \times 2\mu_2$, one grid being shifted by half a cell with respect to the other in a brick-like pattern (inset). Each such grid can further be decomposed into two families of nearly-parallel curves with spacing μ_1 and $2\mu_2$ respectively.

We generate these families of curves over $\bar{\mathcal{S}}$ using the stripe pattern algorithm of Knoppel et al. [KCPS15], which provides local control on stripe spacing and orientation. Since this algorithm requires spacing values per vertex, we compute these values by area-weighted averaging of the per-triangle values given by our flattening algorithm. In practice, we first run the stripe pattern algorithm twice with the spacing fields μ_1 and $2\mu_2$ and the direction fields $\bar{\mathbf{k}}_1$ and $\bar{\mathbf{k}}_2$, respectively, to form one grid of our pattern. The output of the algorithm of Knoppel et al. [KCPS15] is two S^1-valued unit complex fields over $\bar{\mathcal{S}}$, θ and ψ: the zero isolines of $\arg(\theta)$ give the centerlines of the ribbons on one copy of the staggered grid (dark blue lines in inset figure), and the zero isolines of $\arg(\psi)$ (light blue lines) intersect those $\arg(\theta)$-isolines at the ribbon midpoints. We generate the shifted second grid (orange lines) simply by extracting the π isolines of both fields. The curvature κ_1 along these lines (used to define the thicknesses of the associated ribbons) is obtained by linearly interpolating the values of κ_1 at the crossing points between the lines and the edges of the mesh.

The ribbon centerline curves can be directly computed from the staggered grid described above, by cutting the θ-isolines into pieces of length l_r enclosed by the grid intersection points. We clip the ribbons to the boundary of $\bar{\mathcal{S}}$, and we delete ribbons that are shorter than 2mm. Also, we noticed that near singularities, the stripes can deviate significantly from the prescribed direction field to merge or split. We detect these cases by measuring the angle between the ribbon centerline and the direction field, and trim the ribbon when this angle exceeds 25°. Finally, we walk along each ribbon and assign its segments to either the front or back of the fabric depending on the sign of curvature κ_1. In cases where the sign of the curvature changes sign, we split the ribbon in two pieces.

7. Fabrication

Since fabricating our structures requires printing onto fabric that is under uniform, prescribed amount of tension, and requires careful control over the width and thickness of the ribbons being printed, we designed a custom frame structure that attaches to the printer

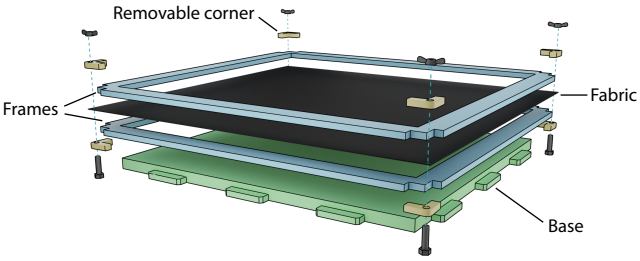


Figure 8: *Our frame structure is composed of two rigid frames, in between which the fabric is clamped in tension. These frames sit on a base that attaches to the printer bed. The four corners of the frames can be removed to be fixed to the sheet of fabric before stretching.*

bed, holds the pre-stretched fabric in place, and allows us to reliably fabricate our results. The frame structure was designed with several goals in mind:

- it should be easy to place the fabric under the prescribed tension, and the frame structure should maintain that tension throughout the printing process;
- it should support the fabric from below to prevent the printer extrusion nozzle from deforming the fabric during printing;
- it should be fixed to the printer bed to prevent sliding during printing;
- it should be easily reversible to print on both sides of fabric, without allowing the pattern on one side to become miscalibrated with respect to that on the other due to sliding or rotation of the fabric.

Figure 8 provides an exploded-view drawing of our design. The structure is composed of two frames that clamp down on the fabric to maintain it under uniform tension. To ease setup of the fabric, the four corners of the frames are removable. We cut a rectangular piece of (unstretched) fabric of the size of the framed scaled by a factor $1/s$, where s is the desired stretching factor. We then detach the four corner pieces from the frame and independently clamp them onto the four corner of the fabric. The corners then snap back into place on the frame, stretching the fabric by the factor s. Once the corners are pulled back to their location on the frames, we fix additional clamps along all sides of the frame to distribute the tension uniformly (see our supplemental video).

The frames sit on a base that is glued to the printer bed. This base has the same size as the inner boundary of the frames, and of the same thickness as one frame (so that the frame snaps into place flush with the base, which ensures that the fabric is supported by the base once the frames are in place). Since the base perfectly fits within the bottom frame, the whole assembly cannot slide. And since the two frames have equal thickness, we can print on both sides of the fabric simply by flipping the frame structure around and placing it back on the base.

We produced all our results with a desktop Ultimaker 2 FDM printer equipped with a large printing nozzle (0.8mm) and configured at low speed (15mm/s), which we found to improve adhesion of plastic on fabric. We used TPU 95A as plastic filament, which is more flexible than standard PLA. We used a finely knitted lycra (80% polyamide, 20% elastane) as fabric, which binds well with the melted plastic and offers enough elasticity to be stretched by a factor $s = 1.6$ in our experiments. This knitted fabric is anisotropic, but we chose to ignore this effect out of simplicity. Notice that rotationally-symmetric designs remain rotationally-symmetric when fabricated (Fig. 1, Fig. 2) and so the effect of fabric anisotropy on the manufactured shape in practice is likely small.

8. Results

Figure 1, Figure 9 and Figure 13 illustrate results produced with our method, including several architectural models and fashion items. We provide turntable animations of these models in the supplemental video, along with a step-by-step demonstration of the fabrication procedure. Note that all these shapes deployed without manual placement of their boundaries, except the ones shown in Figure 12 and Figure 13 for which we used additional support or a textile strengthener, as discussed below.

Our results exhibit various degrees of curvature, including positive and negative curvature achieved by printing on both sides of the fabric (Figure 1) and nearly developable surfaces (*Skirt* in Figure 9). This latter example highlights the benefit of being able to control extrinsic curvature (via the bilayer effect), since developables cannot be fabricated by controlling the intrinsic curvature alone. Fig. 10 plots the curvature values sampled over some of our target shapes, which shows that while the bilayer effect allows us to reproduce nearly-developable surfaces ($\mathbf{k}_2$ close to 0), metric frustration also provides a means to reproduce shapes with moderate positive and negative Gaussian curvature.

We also visualize for each result in Figure 9 its deviation from the target surface: we reconstruct the geometry of the deployed structure from a photogrammetry scan of the fabricated object [Cap], rigidly register it to the target surface $\mathcal{S}$, and for each point on the reconstructed surface, compute the distance to its closest point on $\mathcal{S}$. Errors are expressed as percentages of the bounding box diagonal of $\mathcal{S}$.

Table 1 reports the computation time for each model shown in the paper, along with the average and maximum deviation of the photogrammetry scan from the target surface. Note that we did not scan some models that require additional intervention to support their own weight, as discussed below. While the plastic ribbons produce small-scale relief over the fabricated surface, our method captures the overall shape well with an average deviation of around 1.4%, which corresponds to 1.6 mm on such small-scale objects (11 cm average bounding box diagonal). This level of accuracy is on par with the one of prior methods that exploit similar mechanisms for the lightweight fabrication of deployable structures [POT17; PIC*21]. Table 1 also provides the average deviation from the prescribed direction $\mathbf{k_1}$, as well as the percentage of triangles that exceed the stretching bounds, along with the maximum deviation from the bounds. This deviation remains small for all models, even though some models reach the bounds on a relatively large portion of the surface.

Limitations. The maximum length distortion achievable using our technique, as the surface deploys from the flat to curved state, is

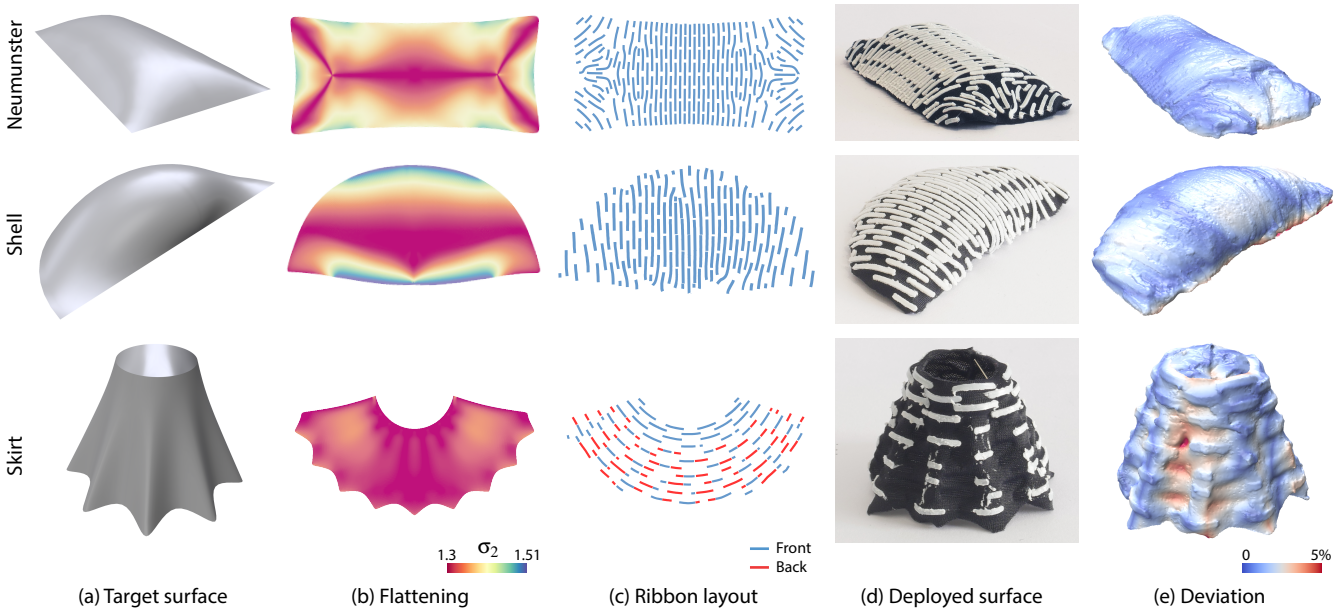


Figure 9: *Representative results produced with our method, including freeform doubly-curved surfaces (*Neumunster *and* Shell*) and a nearly-developable surface achieved by printing on both sides of the fabric (*Skirt*). We visualize the deviation of the scanned fabricated shape from the target surface, expressed as percentages of the bounding box diagonal of the target.*

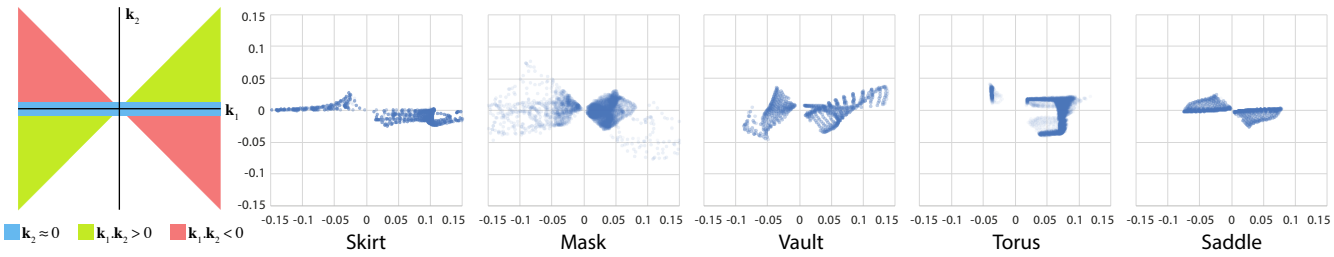


Figure 10: *Visualization of the two principal curvature values sampled over some of our target surfaces. While we expect our method to be most effective at controlling extrinsic curvature where the target shape is nearly-developable (*$\mathbf{k}_2 \approx 0$*, such as throughout the* Skirt*), most of the surfaces we reproduced also exhibit positive and negative Gaussian curvature (*Mask, Vault *and* Torus*), or even solely negative Gaussian curvature (*Octopus Saddle*).*

Table 1: *For each result shown in the paper, we report the computation time, curvature field smoothing* λ*, average and maximum deviation between the scanned fabricated shape and the target (*99^{th} *percentile to remove scanning outliers), average deviation of the ribbon direction from the target direction* $\mathbf{k_1}$*, maximum deviation from the bounds on* σ_1 *and* σ_2*, percentage of triangles that violate the bounds, and average residual energy after convergence. While some shapes exhibit a large portion of triangles outside the bounds, this deviation remains small. The two shapes that have the highest deviation in direction and percentage of triangles out of the bounds are* Octopus saddle *(Fig. 1) and* Vault *(Fig. 13). We note that these two models contain an umbilical region in their center, where the direction field exhibits a singularity.*

Model	**Nb. faces**	**Time** (s)	λ	**Dev. (avg // max)**	$\mathbf{k_1}$ **dev. (avg.** °**)**	**Max dev. (s1 // s2)**	**Bad** σ_1 **//** σ_2 **(%)**	**Avg. energy**
Shell	105792	7.13	0	0.979 // 4.11	0.169	0.004 // 0.002	6.114 // 7.386	0.169
Octopus saddle	49152	0.91	0	2.63 // 8.01	0.731	0.0002 // 0.002	0.191 // 15.448	0.075
Neumunster	147456	5.20	-0.1	0.771 // 3.09	0.058	0.0002 // 0.027	2.100 // 2.740	0.192
Torus	159744	4.31	-0.1	1.14 // 4.96	0.070	0.003 // 0.002	1.127 // 7.699	0.201
Skirt	92160	2.90	-0.01	1.49 // 5.30	0.090	0.008 // 0.003	7.850 // 11.763	0.195
H. A. Center	55296	4.64	0		0.471	0.033 // 0.004	5.619 // 6.104	0.170
Vault	41472	7.53	-0.01		3.777	0.029 // 0.072	28.369 // 16.802	0.077
Mask	67584	1.35	0		0.069	0.007 // 0.003	2.477 // 4.903	0.196

(a) Automatic layout (b) Hand-designed layout

Figure 11: *Limitation. On umbilic surfaces like the sphere, our automatic method aligns the ribbon pattern along an arbitrary direction. This layout offers limited room to flatten the surface while satisfying the stretch bounds, restricting the method to a spherical cap of low curvature (a). Our flattening algorithm can accommodate higher curvature if provided with a hand-designed radial layout (b), but the surface curves too much at its center, possibly due to the singularity of the layout at this location.*

(a) Target surface (b) Without support (c) With support

Figure 12: *Limitation. Without additional support, this architectural model sags under its own weight (b). Adding a cardboard support underneath the structure brings it closer to the target surface (c).*

bounded by the fabric pre-stretch factor (which is $s = 1.6$ for the knitted lycra we used). Consequently, our method cannot reproduce surfaces with patches of large total Gaussian curvature. (One workaround would be to introduce cuts in the parameterization, and stitch the fabric along those cuts to form the surface after printing [SPS18; GMB17].) Although our flattening algorithm is guaranteed to converge, running it on surfaces that cannot be flattened while satisfying the principal stretch bounds will yield a flattening with significant cost function residual (Equation 3): in this case, either the right singular vectors of the resulting mapping J do not align well with the prescribed directions, *i.e.* the flattening does not stretch the surface along the curvature directions; or the singular values of J do not lie within the prescribed bounds and the desired flattening violates the ribbon minimal or maximal spacing constraints.

Our choice of aligning to the direction of principal curvature is particularly effective for nearly-developable surfaces, where the difference between principal curvatures is large. In umbilic regions, the principal curvature direction is not well-defined, and currently our approach chooses one direction arbitrarily in such regions. As a consequence, the ribbon layout is sometimes suboptimal near umbilic points; notice for instance that the *Octopus Saddle* (Fig. 1) and *Vault* (Fig. 13), which feature prominent umbilic regions, also have the largest shape deviation among our examples (Table 1). Fig. 11(a) illustrates failure of our method on a purely umbilic surface, where the arbitrary direction field yields a poor distribution of stretch, to the point where our flattening algorithm only manages to satisfy the bounds for a spherical cap of low curvature, which curves little once fabricated. Fig. 11(b) shows the result obtained with a hand-designed radial direction field, which does yield a flattening that satisfies the distortion bounds for a spherical cap of higher curvature, and a manufactured surface that is positively-curved but oblong (with higher curvature at the north pole than at the equator, possibly due to the direction field singularity at the pole, where ribbons are far from parallel). A potential future improvement would be to relax the curvature-alignment constraint (similar to the soft penalty approach of Panetta et al. [PIC*21]), perhaps weighting alignment by a factor depending on $|\kappa_1 - \kappa_2|$, though the optimization would also need to include terms to prevent shape distortion due to ribbon torsion. Another promising idea for future work is to add anisotropy of the ribbon pattern as an optimization variable, so that the ribbon shape can adjust in response to the ratio $\frac{\kappa_1}{\kappa_2}$ (with ribbons degenerating to disks or squares in umbilic regions, with only their spacing and not orientation encoding intrinsic curvature in those regions, similar to the experiments by Fields et al. [Fie18].)

The small size of the printing area and limited resolution of commodity desktop 3D printers also prevents us from producing large surfaces, or surfaces with geometric features that are finer than can be resolved by our ribbon-based metamaterial. Printing on fabric is also inherently imprecise, as plastic sometimes does not adhere well to fabric, and plastic sometimes leaks from one ribbon to the next as the printing head moves over the surface, as shown in the inset figure. Note however that our algorithm could be applied to larger-scale, more precise fabrication techniques, such as plywood panels glued to a large pre-stretched latex sheet [BAB20], or to higher-end 3D printers with larger print volume or finer resolution.

Finally, our method solves the inverse design task purely geometrically, without simulating the complex interactions between the elastic fabric and the plastic ribbons, nor the effect of external forces like gravity or additional load. Figure 12 and Figure 13 show results on shapes that tend to sag under the effect of gravity, which we corrected for by adding external support or by applying a textile strengthener (Powertex). Nevertheless, our geometric approach also has its strengths, as it achieves a good agreement with the target surface without costly physics-based optimization as part of the design loop. We also note that any simulation-based algorithm would require a good ribbon pattern initialization to converge to a good solution, which our method provides.

9. Conclusion

We introduced a metamaterial composed of a pattern of flexible, incompressible ribbons embedded into a pre-stretched elastic membrane, and described how leveraging the stretch in-between the ribbons and the bilayer effect at the ribbon-fabric interface allows programming the intrinsic and extrinsic curvature of deployable surfaces covered by such a pattern. We also introduced a tailored flattening algorithm that bounds stretch along and across the principal directions of curvature of a target surface, such that lay-

Figure 13: *Architectural model (*Vault*) and fashion items (*Skirt *and* Mask*) prototyped with our method.*

ing out ribbons over the flattened surface yields an accurate reproduction of the target when deployed. Finally, we showed how these contributions allow the rapid inverse design and fabrication of lightweight structures made of plastic ribbons printed on lycra. Potential exciting directions for future research include applying our pattern and the associated algorithms to new kinds of elastic substrates that allow for the creation of larger deployable structures; design of more complex composites with multiple fabric layers; study of how printing n-RoSys or other more complex primitive shapes, beyond stripes, affects the deployed surface geometry; and algorithms for designing structures that deploy into multiple programmed metastable shapes, or that deploy and flatten automatically in response to programmed loads or environmental stimuli.

Acknowledgements

We thank Emilie Yu for help with scanning our fabricated models and editing the supplemental video. We also thank the developers of the open source libraries libigl [JP*18], Geometry Central [SC*19] and Polyscope [Sha*19] that we used for geometry processing and visualization. This work was supported by the ERC Starting Grant D^3 (ERC-2016-STG 714221), the US National Science Foundation (IIS-1910274), and research and software donations from Adobe.

References

[AXZ*18] AHARONI, HILLEL, XIA, YU, ZHANG, XINYUE, et al. "Universal inverse design of surfaces with thin nematic elastomer sheets". *Proceedings of the National Academy of Sciences* 115.28 (2018), 7206–7211. DOI: 10.1073/pnas.1804702115 2, 4, 5.

[BAB20] BERDOS, YORGOS, AGKATHIDIS, ASTERIOS, and BROWN, ANDRE. "Architectural hybrid material composites: computationally enabled techniques to control form generation". *Architectural Science Review* 63.2 (2020), 154–164. DOI: 10.1080/00038628.2019.1666357 3, 11.

[BFS10] BIARD, LUC, FAROUKI, RIDA T., and SZAFRAN, NICOLAS. "Construction of rational surface patches bounded by lines of curvature". *Computer Aided Geometric Design* 27 (5 2010), 359–371. DOI: 10.1016/j.cagd.2010.03.002 4.

[BTM*20] BODDETI, NARASIMHA, TANG, YUNLONG, MAUTE, KURT, et al. "Optimal design and manufacture of variable stiffness laminated continuous fiber reinforced composites". *Scientific reports* 10.1 (Oct. 2020), 1–15. DOI: 10.1038/s41598-020-73333-4 4.

[BvRL*19] BOLEY, J. WILLIAM, van REES, WIM M., LISSANDRELLO, CHARLES, et al. "Shape-shifting structured lattices via multimaterial 4D printing". *Proceedings of the National Academy of Sciences* 116.42 (2019), 20856–20862. DOI: 10.1073/pnas.1908806116 1, 3.

[Cap] CAPTURINGREALITY. *RealityCapture*. https://www.capturingreality.com/. Accessed: 2021-12-09 9.

[CPSP21] CHEN, TIAN, PANETTA, JULIAN, SCHNAUBELT, MAX, and PAULY, MARK. "Bistable Auxetic Surface Structures". *ACM Transactions on Graphics* 40.4 (July 2021), 39:1–39:9. DOI: 10.1145/3450626.3459940 2, 3.

[dCar76] DO CARMO, MANFREDO P. *Differential Geometry of Curves and Surfaces*. Englewood Cliffs, NJ: Prentice-Hall, 1976 4.

[dPel17] DU PELOUX, LIONEL. "Modeling of bending-torsion couplings in active-bending structures : application to the design of elastic gridshells". PhD thesis. Université Paris Est, École des Ponts Paris Tech, Dec. 2017. URL: https://tel.archives-ouvertes.fr/tel-01757782/ 3.

[Fie18] FIELDS, GABE. *Self Forming Structures: An Exploration into 3D Printing on Pre-stretched Fabric*. Nervous System. 2018. URL: https://n-e-r-v-o-u-s.com/blog/?p=8011 3, 11.

[GAE19] GRINIASTY, ITAY, AHARONI, HILLEL, and EFRATI, EFI. "Curved Geometries from Planar Director Fields: Solving the Two-Dimensional Inverse Problem". *Phys. Rev. Lett.* 123 (12 Sept. 2019), 127801. DOI: 10.1103/PhysRevLett.123.127801 3.

[GC16] GUBERAN, CHRISTOPHE and CLOPATH, CARLO. *Active Shoes*. http://www.christopheguberan.ch/active-shoes/. Accessed: 2021-12-02. 2016 2, 3.

[GMB17] GUSEINOV, RUSLAN, MIGUEL, EDER, and BICKEL, BERND. "CurveUps: Shaping Objects from Flat Plates with Tension-Actuated Curvature". *ACM Transactions on Graphics* 36.4 (July 2017), 64:1–64:12. DOI: 10.1145/3072959.3073709 1, 3, 4, 11.

[GMN*16] GLADMAN, A. S., MATSUMOTO, E. A., NUZZO, R. G., et al. "Biomimetic 4D Printing". *Nature Materials* 15 (2016), 413–419. DOI: 10.1038/nmat4544 1, 3.

[GMP*20] GUSEINOV, RUSLAN, MCMAHAN, CONNOR, PÉREZ, JESÚS, et al. "Programming temporal morphing of self-actuated shells". *Nature Communications* 11 (Dec. 2020), 237. DOI: 10.1038/s41467-019-14015-2 3.

[GSD*14] GARG, AKASH, SAGEMAN-FURNAS, ANDREW O., DENG, BAILIN, et al. "Wire Mesh Design". *ACM Transactions on Graphics* 33.4 (July 2014), 66:1–66:12. DOI: 10.1145/2601097.2601106 3.

[HLS07] HORMANN, KAI, LÉVY, BRUNO, and SHEFFER, ALLA. "Mesh Parameterization: Theory and Practice". *ACM SIGGRAPH 2007 Courses*. SIGGRAPH '07. 2007. DOI: 10.1145/1281500.1281510 4.

[IBB15] IARUSSI, EMMANUEL, BOMMES, DAVID, and BOUSSEAU, ADRIEN. "BendFields: Regularized Curvature Fields from Rough Concept Sketches". *ACM Transactions on Graphics* 34.3 (May 2015), 24:1–24:16. DOI: 10.1145/2710026 4.

[IRHS20] ION, ALEXANDRA, RABINOVICH, MICHAEL, HERHOLZ, PHILIPP, and SORKINE-HORNUNG, OLGA. "Shape Approximation by Developable Wrapping". *ACM Transactions on Graphics* 39.6 (Nov. 2020). DOI: 10.1145/3414685.3417835 3.

[JP*18] JACOBSON, ALEC, PANOZZO, DANIELE, et al. *libigl: A simple C++ geometry processing library*. https://libigl.github.io/. 2018 6, 12.

[JSVB21] JOURDAN, DAVID, SKOURAS, MÉLINA, VOUGA, ETIENNE, and BOUSSEAU, ADRIEN. "Printing-on-Fabric Meta-Material for Self-Shaping Architectural Models". *Advances in Architectural Geometry 2020*. Paris, Apr. 2021. URL: https://hal.archives-ouvertes.fr/hal-02925036 3, 4, 7, 8.

[KCD*16] KONAKOVIĆ, MINA, CRANE, KEENAN, DENG, BAILIN, et al. "Beyond Developable: Computational Design and Fabrication with Auxetic Materials". *ACM Transactions on Graphics* 35.4 (July 2016), 89:1–89:11. DOI: 10.1145/2897824.2925944 3, 4.

[KCPS13] KNÖPPEL, FELIX, CRANE, KEENAN, PINKALL, ULRICH, and SCHRÖDER, PETER. "Globally optimal direction fields". *ACM Transactions on Graphics* 32.4 (July 2013), 59:1–59:10. DOI: 10.1145/2461912.2462005 2, 6, 7.

[KCPS15] KNÖPPEL, FELIX, CRANE, KEENAN, PINKALL, ULRICH, and SCHRÖDER, PETER. "Stripe Patterns on Surfaces". *ACM Transactions on Graphics* 34 (4 July 2015), 39:1–39:11. DOI: 10.1145/2767000 2, 4, 8.

[KHB*12] KIM, JUNGWOOK, HANNA, JAMES A., BYUN, MYUNGHWAN, et al. "Designing Responsive Buckled Surfaces by Halftone Gel Lithography". *Science* 335.6073 (2012), 1201–1205. DOI: 10.1126/science.1215309 3.

[KMM17] KILIAN, MARTIN, MONSZPART, ARON, and MITRA, NILOY J. "String Actuated Curved Folded Surfaces". *ACM Transactions on Graphics* 36.3 (May 2017), 25:1–25:13. DOI: 10.1145/3015460 3.

[KPCP18] KONAKOVIĆ-LUKOVIĆ, MINA, PANETTA, JULIAN, CRANE, KEENAN, and PAULY, MARK. "Rapid Deployment of Curved Surfaces via Programmable Auxetics". *ACM Transactions on Graphics* 37.4 (July 2018), 106:1–106:13. DOI: 10.1145/3197517.3201373 1, 2.

[Kyc19] KYCIA, AGATA. "Hybrid Textile Structures as Means of Material-informed Design Strategy". *CA²RE Berlin Proceedings: Conference for Artistic and Architectural (Doctoral) Research*. Berlin, 2019, 34–35 3.

[LZX*08] LIU, LIGANG, ZHANG, LEI, XU, YIN, et al. "A Local/Global Approach to Mesh Parameterization". *Proceedings of the Symposium on Geometry Processing*. SGP '08. Copenhagen, 2008, 1495–1504. DOI: 10.5555/1731309.1731336 2, 4, 6.

[MLB16] MIGUEL, EDER, LEPOUTRE, MATHIAS, and BICKEL, BERND. "Computational Design of Stable Planar-Rod Structures". *ACM Transactions on Graphics* 35.4 (July 2016), 86:1–86:11. DOI: 10.1145/2897824.2925978 4.

[MPI*18] MALOMO, LUIGI, PÉREZ, JESÚS, IARUSSI, EMMANUEL, et al. "FlexMaps: Computational Design of Flat Flexible Shells for Shaping 3D Objects". *ACM Transactions on Graphics* 37.6 (Dec. 2018), 241:1–241:14. DOI: 10.1145/3272127.3275076 3.

[MWS*20] MA, ZHAO, WALZER, ALEXANDER, SCHUMACHER, CHRISTIAN, et al. "Designing Robotically-Constructed Metal Frame Structures". *Computer Graphics Forum* 39.2 (2020), 411–422. DOI: 10.1111/cgf.13940 4.

[NAH*18] NARAYANAN, VIDYA, ALBAUGH, LEA, HODGINS, JESSICA, et al. "Automatic Machine Knitting of 3D Meshes". *ACM Transactions on Graphics* 37.3 (Aug. 2018). DOI: 10.1145/3186265 4.

[NBB*16] NA, JUN-HEE, BENDE, NAKUL P., BAE, JINHYE, et al. "Grayscale gel lithography for programmed buckling of non-Euclidean hydrogel plates". *Soft Matter* 12 (22 2016), 4985–4990. DOI: 10.1039/C6SM00714G 3.

[OR07] OXMAN, NERI and ROSENBERG, JESSE LOUIS. "Material-based Design Computation An Inquiry into Digital Simulation of Physical Material Properties as Design Generators". *International Journal of Architectural Computing* 5.1 (2007), 25–44. DOI: 10.1260/147807707780912985 3.

[PIC*21] PANETTA, JULIAN, ISVORANU, FLORIN, CHEN, TIAN, et al. "Computational Inverse Design of Surface-Based Inflatables". *ACM Transactions on Graphics* 40.4 (July 2021), 40:1–40:14. DOI: 10.1145/3450626.3459789 1–5, 9, 11.

[PKI*19] PANETTA, J., KONAKOVIĆ-LUKOVIĆ, M., ISVORANU, F., et al. "X-Shells: A New Class of Deployable Beam Structures". *ACM Transactions on Graphics* 38.4 (July 2019), 83:1–83:15. DOI: 10.1145/3306346.3323040 1, 3.

[PLBM20] PILLWEIN, STEFAN, LEIMER, KURT, BIRSAK, MICHAEL, and MUSIALSKI, PRZEMYSLAW. "On Elastic Geodesic Grids and Their Planar to Spatial Deployment". *ACM Transactions on Graphics* 39.4 (July 2020), 125:1–125:12. DOI: 10.1145/3386569.3392490 3.

[POT17] PÉREZ, JESÚS, OTADUY, MIGUEL A., and THOMASZEWSKI, BERNHARD. "Computational Design and Automated Fabrication of Kirchhoff-Plateau Surfaces". *ACM Transactions on Graphics* 36.4 (July 2017), 62:1–62:12. DOI: 10.1145/3072959.3073695 2, 3, 7, 9.

[PSJH17] PEZZULLA, MATTEO, STOOP, NORBERT, JIANG, XIN, and HOLMES, DOUGLAS. "Curvature-driven morphing of non-Euclidean shells". *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 473.2201 (May 2017). DOI: 10.1098/rspa.2017.0087 3.

[RHS17] RABINOVICH, MICHAEL, HOFFMANN, TIM, and SORKINE-HORNUNG, OLGA. "Discrete Geodesic Nets for Modeling Developable Surfaces". *ACM Transactions on Graphics* 37 (July 2017). DOI: 10.1145/3180494 3.

[RPC*21] REN, YINGYING, PANETTA, JULIAN, CHEN, TIAN, et al. "3D Weaving with Curved Ribbons". *ACM Transactions on Graphics* 40.4 (July 2021), 127:1–127:15. DOI: 10.1145/3450626.3459788 4.

[RPPS17] RABINOVICH, MICHAEL, PORANNE, ROI, PANOZZO, DANIELE, and SORKINE-HORNUNG, OLGA. "Scalable Locally Injective Mappings". *ACM Transactions on Graphics* 36.2 (2017). DOI: 10.1145/2983621 6.

[SC*19] SHARP, NICHOLAS, CRANE, KEENAN, et al. *geometry-central*. www.geometry-central.net. 2019 12.

[SC17] SAWHNEY, ROHAN and CRANE, KEENAN. "Boundary First Flattening". *ACM Transactions on Graphics* 37.1 (Dec. 2017), 5:1–5:14. DOI: 10.1145/3132705 4.

[SC18] SHARP, NICHOLAS and CRANE, KEENAN. "Variational Surface Cutting". *ACM Transactions on Graphics* 37.4 (July 2018), 156:1–156:13. DOI: 10.1145/3197517.3201356 3.

[SCBV19] SAGEMAN-FURNAS, ANDREW O., CHERN, ALBERT, BEN-CHEN, MIRELA, and VAXMAN, AMIR. "Chebyshev Nets from Commuting PolyVector Fields". *ACM Transactions on Graphics* 38.6 (Nov. 2019), 172:1–172:16. DOI: 10.1145/3355089.3356564 4.

[SE10] SHARON, ERAN and EFRATI, EFI. "The mechanics of non-Euclidean plates". *Soft Matter* 6 (22 2010), 5693–5704. DOI: 10.1039/C0SM00479K 3.

[SGC18] STEIN, ODED, GRINSPUN, EITAN, and CRANE, KEENAN. "Developability of Triangle Meshes". *ACM Transactions on Graphics* 37.4 (July 2018), 77:1–77:14. DOI: 10.1145/3197517.3201303 3.

[Sha*19] SHARP, NICHOLAS et al. *Polyscope*. www.polyscope.run. 2019 12.

[SLMB05] SHEFFER, ALLA, LÉVY, BRUNO, MOGILNITSKY, MAXIM, and BOGOMYAKOV, ALEXANDER. "ABF++: Fast and Robust Angle Based Flattening". *ACM Transactions on Graphics* 24.2 (Apr. 2005), 311–330. DOI: 10.1145/1061347.1061354 4.

[SPS18] SCHÜLLER, CHRISTIAN, PORANNE, ROI, and SORKINE-HORNUNG, OLGA. "Shape Representation by Zippables". *ACM Transactions on Graphics* 37.4 (July 2018), 78:1–78:13. DOI: 10.1145/3197517.3201347 11.

[TBWP16] TANG, CHENGCHENG, BO, PENGBO, WALLNER, JOHANNES, and POTTMANN, HELMUT. "Interactive Design of Developable Surfaces". *ACM Transactions on Graphics* 35.2 (Jan. 2016), 12:1–12:12. DOI: 10.1145/2832906 3.

[TTZ*20] TRICARD, THIBAULT, TAVERNIER, VINCENT, ZANNI, CÉDRIC, et al. "Freely Orientable Microstructures for Designing Deformable 3D Prints". *ACM Transactions on Graphics* 39.6 (Nov. 2020), 211:1–211:16. DOI: 10.1145/3414685.3417790 4.

[VCD*16] VAXMAN, AMIR, CAMPEN, MARCEL, DIAMANTI, OLGA, et al. "Directional Field Synthesis, Design, and Processing". *Computer Graphics Forum* 35 (2016), 545–572. DOI: 10.1111/cgf.12864 4.

[VZF*19] VEKHTER, JOSH, ZHUO, JIACHENG, FANDINO, LUISA F GIL, et al. "Weaving Geodesic Foliations". *ACM Transactions on Graphics* 38.4 (July 2019), 34:1–34:22. DOI: 10.1145/3306346.3323043 4.

[ZSGS04] ZHOU, KUN, SYNDER, JOHN, GUO, BAINING, and SHUM, HEUNG-YEUNG. "Iso-Charts: Stretch-Driven Mesh Parameterization Using Spectral Analysis". *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*. SGP '04. Nice, 2004, 45–54. DOI: 10.1145/1057432.1057439 4.

Appendix A: Optimal stretching matrices Σ^e

We obtain the per element stretching matrix $\Sigma^e = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}$ by solving the problem

$$(\tilde{\sigma}_1^e, \tilde{\sigma}_2^e) = \underset{\substack{(\sigma_1,\sigma_2) \\ \sigma_1^{\min} \le \sigma_1 \le \sigma_1^{\max}, \\ \sigma_2^{\min} \le \sigma_2 \le \sigma_2^{\max}}}{\operatorname{argmin}} \underbrace{\|J^e(\bar{\mathbf{x}}) - R^e_{\bar{\mathcal{S}}} \Sigma^e R^e_\theta\|_F^2}_{E(\sigma_1,\sigma_2)}. \tag{11}$$

Noting that the Frobenius norm of any matrix A is invariant under transposition and rotation of the matrix, i.e. $\|A^t\|_F^2 = \|A\|_F^2$ and $\|RA\|_F^2 = \|AR\|_F^2 = \|A\|_F^2$ for any rotation matrix R, we rewrite the cost function in Equation (11) as

$$\begin{aligned} E(\sigma_1,\sigma_2) &= \|(R^e_{\bar{\mathcal{S}}})^t (J^e - R^e_{\bar{\mathcal{S}}} \Sigma^e R^e_\theta)(R^e_\theta)^t\|_F^2 \\ &= \|(R^e_{\bar{\mathcal{S}}})^t J^e (R^e_\theta)^t - \Sigma^e\|_F^2 \\ &= \|R^e_\theta (J^e)^t R^e_{\bar{\mathcal{S}}} - \Sigma^e\|_F^2. \end{aligned} \tag{12}$$

Expanding the right term gives us

$$\begin{aligned} E(\sigma_1,\sigma_2) &= \operatorname{tr}[(R^e_\theta (J^e)^t R^e_{\bar{\mathcal{S}}} - \Sigma^e)^t (R^e_\theta (J^e)^t R^e_{\bar{\mathcal{S}}} - \Sigma^e)] \\ &= \operatorname{tr}[(R^e_\theta (J^e)^t R^e_{\bar{\mathcal{S}}})^t R^e_\theta (J^e)^t R^e_{\bar{\mathcal{S}}}] - 2\operatorname{tr}[\Sigma^e R^e_\theta (J^e)^t R^e_{\bar{\mathcal{S}}}] \\ &\quad + \operatorname{tr}[(\Sigma^e)^t \Sigma^e] \\ &= \operatorname{tr}[(R^e_\theta (J^e)^t R^e_{\bar{\mathcal{S}}})^t R^e_\theta (J^e)^t R^e_{\bar{\mathcal{S}}}] - 2\sigma_1^2 [R^e_\theta (J^e)^t R^e_{\bar{\mathcal{S}}}]_{11} \\ &\quad - 2\sigma_2^2 [R^e_\theta (J^e)^t R^e_{\bar{\mathcal{S}}}]_{22} + \sigma_1^2 + \sigma_2^2, \end{aligned} \tag{13}$$

where $\operatorname{tr}[A]$ denotes the trace of the matrix A, and $[R^e_\theta (J^e)^t R^e_{\bar{\mathcal{S}}}]_{11}$ and $[R^e_\theta (J^e)^t R^e_{\bar{\mathcal{S}}}]_{22}$ are the two diagonal terms of $R^e_\theta (J^e)^t R^e_{\bar{\mathcal{S}}}$.

This is a quadratic function in σ_1 and σ_2 whose minimum subject to $\sigma_i^{\min} \le \sigma_i \le \sigma_i^{\max}$ is given by

$$\tilde{\sigma}_i^e = \begin{cases} \sigma_i^{\min} & \text{if } \hat{\sigma}_i < \sigma_i^{\min}, \\ \hat{\sigma}_i & \text{if } \sigma_i^{\min} \le \hat{\sigma}_i \le \sigma_i^{\max}, \\ \sigma_i^{\max} & \text{if } \sigma_i^{\max} < \hat{\sigma}_i, \end{cases} \quad i = \{1,2\}, \tag{14}$$

where $\hat{\sigma}_i$ is solution of $\frac{\partial E}{\partial \sigma_i}(\sigma_1,\sigma_2) = 0$, i.e. $\hat{\sigma}_i = [R^e_\theta (J_e)^t R^e_{\bar{\mathcal{S}}}]_{ii}$.

Appendix B: Optimal rotation matrices $R^e_{\bar{\mathcal{S}}}$

We obtain the per element rotation matrix $R^e_{\bar{\mathcal{S}}}$ by solving the orthogonal Procrustes problem

$$\tilde{R}^e_{\bar{\mathcal{S}}} = \underset{R^e_{\bar{\mathcal{S}}} \in SO(2)}{\operatorname{argmin}} \left\|J^e - R^e_{\bar{\mathcal{S}}} \Sigma^e R^e_\theta\right\|_F^2. \tag{15}$$

Expanding the right hand side of equation (15) gives us

$$\begin{aligned} \left\|J^e - R^e_{\bar{\mathcal{S}}} \Sigma^e R^e_\theta\right\|_F^2 &= \operatorname{tr}\left((J^e - R^e_{\bar{\mathcal{S}}} \Sigma^e R^e_\theta)^t (J^e - R^e_{\bar{\mathcal{S}}} \Sigma^e R^e_\theta)\right) \\ &= \operatorname{tr}\left((J^e)^t J^e\right) - 2\operatorname{tr}\left((J^e)^t R^e_{\bar{\mathcal{S}}} \Sigma^e R^e_\theta\right) \\ &\quad + \operatorname{tr}\left((R^e_{\bar{\mathcal{S}}} \Sigma^e R^e_\theta)^t R^e_{\bar{\mathcal{S}}} \Sigma^e R^e_\theta\right). \end{aligned} \tag{16}$$

Terms that do not depend on $R^e_{\bar{\mathcal{S}}}$ do not change the minimum of the function (16) with respect to $R^e_{\bar{\mathcal{S}}}$ and can be discarded so that we can write

$$\begin{aligned} \tilde{R}^e_{\bar{\mathcal{S}}} &= \underset{R^e_{\bar{\mathcal{S}}} \in SO(2)}{\operatorname{argmax}} \operatorname{tr}\left((J^e)^t R^e_{\bar{\mathcal{S}}} \Sigma^e R^e_\theta\right) \\ &= \underset{R^e_{\bar{\mathcal{S}}} \in SO(2)}{\operatorname{argmax}} \operatorname{tr}\left(R^e_{\bar{\mathcal{S}}} \Sigma^e R^e_\theta (J^e)^t\right) \\ &= UV^t, \end{aligned} \tag{17}$$

where U and V are the orthogonal matrices of the singular value decomposition of the matrix $J^e (R^e_\theta)^t \Sigma^e$ up to flipping the sign of the column of U corresponding to the smallest singular value so that $\det(UV^t) > 0$. Indeed, in that case, $J^e (R^e_\theta)^t \Sigma^e = USV^t$ with $S = \begin{pmatrix} S_1 & 0 \\ 0 & S_2 \end{pmatrix}$ the diagonal matrix containing the singular values of $J^e (R^e_\theta)^t \Sigma^e$ and $UV^t \in SO(2)$. This allows us to write

$$\begin{aligned} \operatorname{tr}\left(R^e_{\bar{\mathcal{S}}} \Sigma^e R^e_\theta (J^e)^t\right) &= \operatorname{tr}\left(R^e_{\bar{\mathcal{S}}} VSU^t\right) \\ &= \operatorname{tr}\left(U^t R^e_{\bar{\mathcal{S}}} VS\right) \\ &= S_1 [H]_{11} + S_2 [H]_{22}, \ H = U^t R^e_{\bar{\mathcal{S}}} V. \end{aligned} \tag{18}$$

Since $H \in SO(2)$, $[H]_{11}$ and $[H]_{22}$ are of the form $[H]_{11} = [H]_{22} = \cos\theta$, $\theta \in \mathbb{R}$. The maximum of $\operatorname{tr}\left(R^e_{\bar{\mathcal{S}}} \Sigma^e R^e_\theta (J^e)^t\right)$ is thus reached when $H = I$, i.e. $R^e_{\bar{\mathcal{S}}} = UV^t$.