Shrink & Morph: 3D-printed self-shaping shells actuated by a shape memory effect

David Jourdan, Pierre-Alexandre Hugron, Camille Schreck, Jonas Martinez, Sylvain Lefebvre

Shrink & Morph: 3D-printed self-shaping shells actuated by a shape memory effect

DAVID JOURDAN, Université de Lorraine, CNRS, and Inria, France
PIERRE-ALEXANDRE HUGRON, Université de Lorraine, CNRS, and Inria, France
CAMILLE SCHRECK, Université de Lorraine, CNRS, and Inria, France
JONÀS MARTÍNEZ, Université de Lorraine, CNRS, and Inria, France
SYLVAIN LEFEBVRE, Université de Lorraine, CNRS, and Inria, France

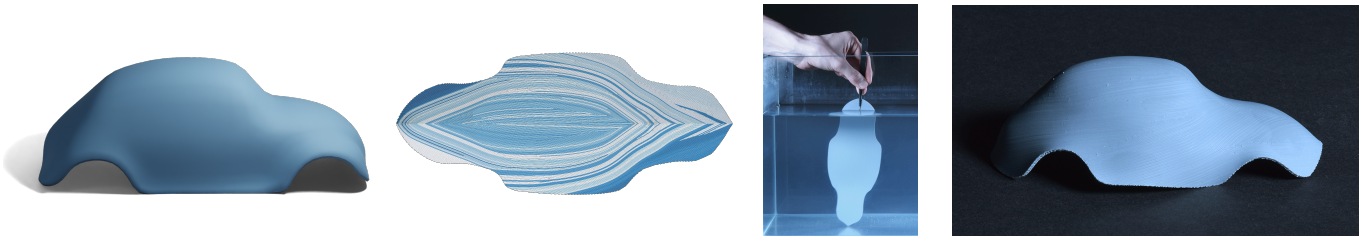


Fig. 1. Given a target surface (left), our method optimizes a dense set of deposition trajectories that form a printed plate. Once the plate is immersed in hot water, it deforms and closely approaches the target surface (right).

While 3D printing enables the customization and home fabrication of a wide range of shapes, fabricating freeform thin-shells remains challenging. As layers misalign with the curvature, they incur structural deficiencies, while the curved shells require large support structures, typically using more material than the part itself.

We present a computational framework for optimizing the internal structure of 3D printed plates such that they morph into a desired freeform shell when heated. This exploits the shrinkage effect of thermoplastics such as PLA, which store internal stresses along the deposition directions. These stresses get released when the material is heated again above its glass transition temperature, causing an anisotropic deformation that induces curvature.

Our inverse design method takes as input a freeform surface and finds an optimized set of deposition trajectories in each layer such that their anisotropic shrinkage deforms the plate into the prescribed surface geometry. We optimize for a continuous vector field that varies across the plate and within its thickness. The algorithm then extracts a set of deposition trajectories from the vector field in order to fabricate the flat plates on standard FFF printers. We validate our algorithm on freeform, doubly-curved surfaces.

CCS Concepts: • **Computing methodologies** → **Computer graphics**; • **Applied computing** → **Computer-aided design**.

Additional Key Words and Phrases: Additive Fabrication, Deployable Structures, Shape Memory Polymers.

Authors' addresses: David Jourdan, david.jourdan@inria.fr, Université de Lorraine, CNRS, and Inria, LORIA, France; Pierre-Alexandre Hugron, pierre-alexandre.hugron@inria.fr, Université de Lorraine, CNRS, and Inria, LORIA, France; Camille Schreck, camille.schreck@inria.fr, Université de Lorraine, CNRS, and Inria, LORIA, France; Jonàs Martínez, jonas.martinez-bayona@inria.fr, Université de Lorraine, CNRS, and Inria, LORIA, France; Sylvain Lefebvre, sylvain.lefebvre@inria.fr, Université de Lorraine, CNRS, and Inria, LORIA, 615 Rue du Jardin-Botanique, Villers-lès-Nancy, 54506, France.

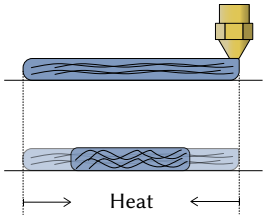
ACM Reference Format:
David Jourdan, Pierre-Alexandre Hugron, Camille Schreck, Jonàs Martínez, and Sylvain Lefebvre. 2023. Shrink & Morph: 3D-printed self-shaping shells actuated by a shape memory effect. *ACM Trans. Graph.* 42, 6, Article 187 (December 2023), 13 pages. https://doi.org/10.1145/3618386

1 INTRODUCTION

Freeform thin shell surfaces appeal to both designers and engineers for their aesthetics and performance. They are commonly used to form the outer shell of home appliances and handheld devices. However, they involve a significantly more complex and costly fabrication process than their flat or single-curved counterparts.

In particular, the additive manufacturing of thin shells usually requires large volumes of supports, while misalignment between shell curvature and printing layers leads to staircase artifacts and poor structural performance (see Fig. 2).

One approach to circumvent these limitations is to consider *self-shaping surfaces*: plates fabricated flat that deform into a specific geometry in response to different stimuli such as heat, pH, or moisture. These approaches avoid the aforementioned issues, show great potential for reducing fabrication costs, and allow postponing deployment until on-site placement.



In this work, we fabricate self-shaping surfaces with Fused Filament Fabrication (FFF), one of the least expensive and most widely used 3D printing techniques. We propose to use the shape memory effect of thermoplastics such as polylactic acid (PLA): polymer chains in the material tend to align along deposition trajectories when printed, their trajectory-aligned distribution turns to a more

isotropic state when the material is reheated above its so-called *glass transition temperature* $T_g \approx 60\,^{\circ}\mathrm{C}$ causing the thermoplastic material to shrink along the printing direction. Our experiments verify and investigate further this physical anisotropic property, which is crucial to our method. As detailed in Section 2, studies have shown that parameters such as speed, layer thickness, and printing temperature can influence the amount of length change induced. However, these parameters cannot be easily varied locally within the plate, and understanding their effect on the shrinking behavior requires precise calibration procedures.

Instead, we propose to optimize the *geometry of the printing trajectories*: by varying the local direction of the trajectories, we control the *direction of the anisotropic deformation* (see Figure 1). However, controlling the in-plane deformation is insufficient, as the same 2D deformation map may result in different 3D shapes. Conversely, curved shapes can exist without in-plane deformations, e.g., developable surfaces.

To gain more control over curvature, we propose to vary the deformation direction not only in-plane but also in the normal direction – the plate thickness – by having printing trajectories with different orientations from layer to layer.

We devise an inverse design method that takes as input a freeform surface and outputs an optimized set of deposition trajectories, such that when the fabricated plate is immersed in hot water, it deforms into the prescribed surface geometry. We optimize a continuous vector field that varies within the plate thickness to obtain freeform surfaces with non-trivial curvature. After optimization, the algorithm extracts a set of deposition trajectories from the vector field in order to fabricate the flat plates. Remarkably, by tailoring the path orientation along the plate layers, our method can control both Gaussian and normal curvature using a single base material.

Contributions. In summary, we introduce:

- An end-to-end framework for 3D printing self-shaping thin-shells, capable of reproducing freeform surfaces. This is achieved through the sole control of the geometry of the printing trajectories within and across layers.
- A novel formulation for morphing shape-memory polymers based on the theory of non-Euclidean plates that allows for variation in the normal direction of the plate to control both Gaussian and normal curvature. This takes into account specificities of 3D printing, in particular the gradation of shrinkage ratios between top and bottom layers.
- A precise calibration procedure based on a study on the shape-memory properties of PLA with respect to parameters such as heat, layer height, and printing speed.

An open-source implementation of our method is available at https://github.com/DavidJourdan/shrink-morph

2 RELATED WORK

We call *self-shaping surfaces* initially flat plates that deform into curved 3D surfaces without any external actuation forces. Such surfaces can result from natural processes such as growth or drying processes or can be manufactured to react to a given stimulus that will drive the deformation.

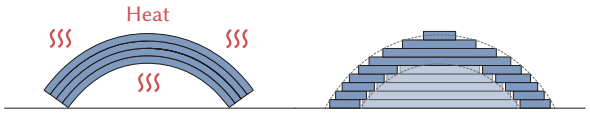


Fig. 2. Layer by layer fabrication (right) leads to staircase artifacts and requires support, while our method (left) enables to have layers aligned with the surface and does not need support.

Manufactured self-shaping surfaces come in a variety of forms and materials. Examples include hydrogels [Gladman et al. 2016; Kim et al. 2012; Na et al. 2016] and nematic elastomers [Aharoni et al. 2018] which can expand when heated or exposed to water, hygroscopic wood-based materials [Cheng et al. 2021; Grönquist et al. 2019], pre-tensioned membranes combined with rigid elements [Guseinov et al. 2020, 2017; Jourdan et al. 2022; Pérez et al. 2017] and more exotic materials such as pasta which self-forms when cooked [Tao et al. 2019, 2021; Wang et al. 2017].

Self-shaping PLA structures. Among these possible techniques, exploiting the shape-memory properties of PLA in fused filament fabrication presents the advantage of being easily accessible through the widely available FFF 3D printers. This leads to direct applications for educational, hobbyist, and repair projects, as well as ongoing research in 4D printing and robotics.

Many approaches rely on fabricating discrete elements that behave as actuators. These elements use a predefined trajectory pattern producing a known, calibrated behavior. The elements are then instantiated at specific positions and orientations to guide the deformation of the surrounding passive parts. Some techniques rely directly on shrinkage [Chalissery et al. 2022; Moon et al. 2022; Wang et al. 2018] while others rely on a *bilayer effect* to induce curvature [van Manen et al. 2017; Wang et al. 2019; Yu et al. 2020]: different predefined patterns with different shrinking behaviors are used for the top and bottom layers. For instance, An et al. [2018] uses two different materials – shrinking PLA and non-shrinking TPU – while Sun et al. [2021] only heats specific parts using conductive PLA. As opposed to discrete elements, we explore a controlled, continuous deformation throughout the plate to obtain thin curved shells.

The Geodesy approach [Gu et al. 2019, 2020] is closest to our objectives but is, however, limited to height-fields as the deposition trajectories always align with the isolines of the input field. In addition, the contraction rate is controlled by continuously varying layer thicknesses. This necessitates a material-specific calibration to precisely quantify the relationship between layer height and shrinking ability. This makes the method difficult to implement on commercial PLA filaments, which exhibit a wide range of behaviors. Although our method also needs some calibration, we only need to find the shrinking ratio for one set of printing parameters.

Deshpande et al. [2022] experiment with controlling the orientation of the deposition trajectories. However, this is not incorporated into an inverse design method. This work demonstrates that heating elements can be locally embedded inside the surface to locally trigger shrinkage.

To the best of our knowledge, our method is the first to optimize for the orientation of trajectories across layers in order to achieve a prescribed, continuous deformation.

Thermomechanical studies of shape memory PLA. The shape memory effect of PLA is a complex phenomenon that depends on various factors. Several material science researchers studied the influence of different parameters on the deformation, such as nozzle temperature [Chalissery et al. 2022], build plate temperature [Kačergis et al. 2019], printing speed [Kačergis et al. 2019; Rajkumar and Shanmugam 2018; van Manen et al. 2017; Zhang et al. 2015], thickness of the plate [Rajkumar and Shanmugam 2018] or the flow of material [van Manen et al. 2017].

As the experiments depend on numerous factors, the results are not always consistent across the literature. Our own experiments showed that results may vary widely depending on the specific PLA filament used, with variations across vendors and colors. This is likely due to the influence of specific additives and pigments, whose exact composition is undisclosed. We provide the parameters used for our examples on specific, commonly available brands of PLA and detail in Section 4 our own experiments to find print parameters maximizing the shrinking performance.

Anisotropic parameterization for deployable structures. The flattening operation required for the inverse design problem to go from the curved target surface to the flat plate to be printed can be represented as a parameterization of a surface embedded in $\mathbb{R}^3$ onto a flat, 2D domain. When materials are deployed from locally *isotropic* deformations – for instance, with auxetic materials – conformal maps have been shown to be the proper framework to geometrically capture their deformation [Konaković et al. 2016] and to build inverse design tools [Chen et al. 2021; Konaković-Luković et al. 2018].

When materials morph from *anisotropic* deformation fields, however, conformal maps cannot be used, and one has to resort to different formalisms. For instance, Griniasty et al. [2019] employed hyperbolic PDE solvers to solve the inverse design problem for nematic elastomers. This technique is not readily applicable to our case as it does not model a change of anisotropy across layers required to account for extrinsic curvature.

Another strategy used for the inverse design of nematic elastomers is to use a local-global solver to find a parameterization much in the same way as ARAP parameterization [Aharoni et al. 2018]. Panetta et al. [2021] use this local-global solver to initialize their non-linear optimizer to design inflatable plates with oriented channels triggering curvature. We employ a similar optimization strategy with a very different simulation model: to account for the fact that the channels are much easier to bend across the air tubes than along, they add a custom term in their optimization, whereas our simulation model for anisotropic, non-euclidean plates allows us to find the proper scaling of the bending terms without introducing an additional free parameter.

Curved printing, conformal slicing. A different take on the fabrication of shells is to rely on curved printing [Zhang et al. 2022]. In particular, Hong et al. [2022] propose a method to 3D print shells using conformal slicing: a 5-axis printer prints a curved surface on top of a PVA support structure that dissolves in water. This, of course, requires a custom 3D printer with 5-axis capabilities and does not remove the need for large volumes of support (compared to the fabricated shell volume).

3 OUR APPROACH

An overview of our inverse design approach is given in Figure 3. Our method takes as input a triangle mesh embedded in $\mathbb{R}^3$ representing the desired target (deployed) surface shape and computes (1) a flattening of the surface into the plane, (2) a three-dimensional vector field covering the plane and varying across its thickness, and (3) outputs a set of deposition trajectories following the vector field for fabrication. We give below a brief overview of each step.

Flattening. We first obtain a planar parameterization of the input surface, which is suitable for optimizing the trajectory orientations. We use a local/global iterative algorithm akin to an As-Rigid-As-Possible parameterization [Liu et al. 2008]. A similar approach was used for nematic elastomers and inflatable structures [Aharoni et al. 2018; Panetta et al. 2021].

Vector field modeling. We solve for a vector field represented as plane angles θ throughout the plate's volume. However, the plate being thin – its thickness much smaller than its other dimensions – we use two angle fields θ_1 and θ_2 to model it with a linear variation across the thickness: $\theta(x, y, z) = \theta_1(x, y) + \frac{z}{h}\theta_2(x, y), z \in [-\frac{h}{2}, \frac{h}{2}]$ where h is the plate thickness.

Vector field optimization. The flattening procedure gives us an initial guess for the vector field, which is used to initialize our inverse optimization method determining optimized printing orientations. In particular, we obtain the value of θ_1 from this flattening procedure.

The heart of our technique is an accurate *forward simulation* method that determines the shape that would be obtained given the vector field. We derive it from the theory of non-Euclidean plates [Efrati et al. 2009] as detailed in Section 3.1.

We use the forward approach to find an equilibrium shape closest to the target, using the adjoint method from sensitivity analysis to optimize for the vector field. We detail the inverse optimization in Section 3.2. A key novelty of our technique is to model the anisotropic bending in the optimization process. This is essential to capture the material behavior adequately.

Trajectory extraction. In each fabrication layer, we generate equally spaced trajectories aligned with the optimized orientations. We rely on the Stripe Patterns algorithm [Knöppel et al. 2015]. The entire process is described in Section 3.2.3.

3.1 Forward problem

In order to tackle the inverse optimization, we first have to solve the forward problem: given trajectory orientations in a flat 3D printed plate, we determine its final configuration after re-heating above the glass transition temperature T_g.

More precisely, given an initial flat configuration $\hat{\mathcal{S}} \in \mathbb{R}^2$ and a scalar field θ representing the orientations, we have to simulate how lengths on the surface change locally so as to obtain the final curved shape. This problem is reminiscent of similar phenomena

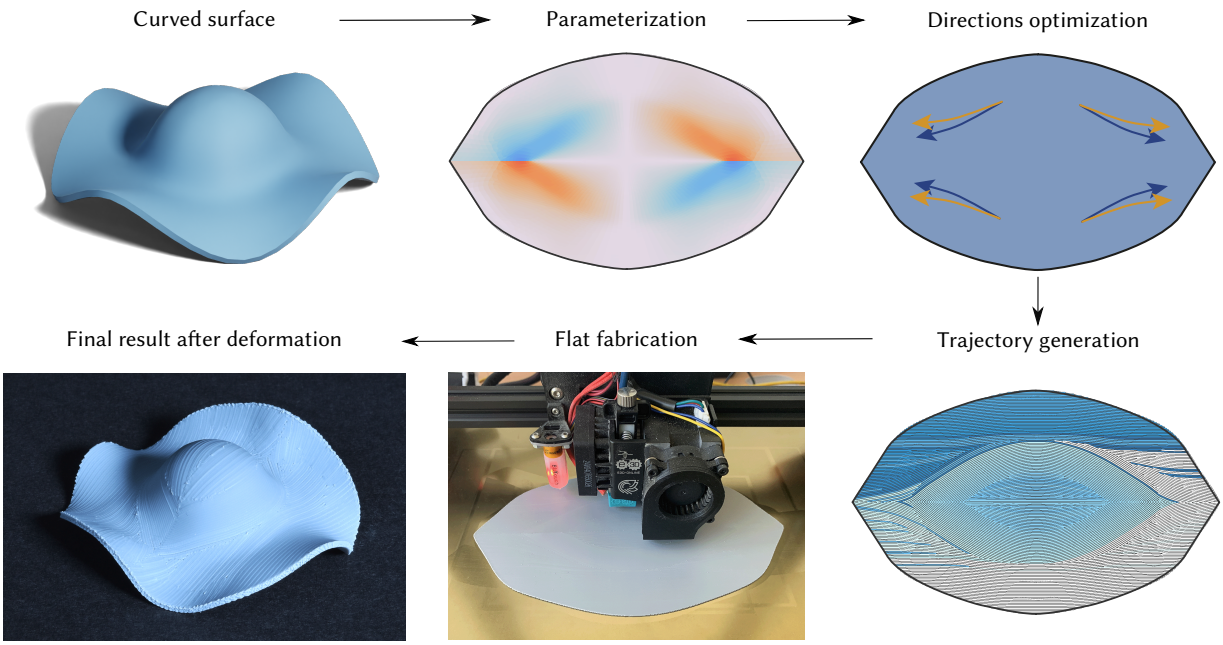


Fig. 3. Overview of our inverse design method. From an input surface, our system generates a 2D parameterization and a set of direction fields for each layer of a corresponding flat plate. Trajectories are extracted along the direction field, allowing for fabrication in PLA material on an extrusion 3D printer. Once dipped into hot water, the plate deforms into the desired surface.

where flat sheets deform under environmental changes (e.g., drying leaves, moist paper, etc. see [Chen et al. 2018]). These phenomena can be approximated geometrically as a metric change within the rest configuration of the material. This initially flat plate therefore gets imbued with a reference configuration whose metric may represent a surface that is not embeddable in $\mathbb{R}^3$, which is why we refer to these structures as *non-Euclidean plates* [Efrati et al. 2009; Pezzulla et al. 2017]. We begin by describing in Section 3.1.1 a general elastic energy formulation (often attributed to W.T. Koiter) whose derivations can be found in more detail in [Efrati et al. 2009]. We explain how to adapt this model for anisotropic materials and derive the exact quantities specific to our problem in Section 3.1.2. Finally, we solve the problem using a discretization similar to [Sperl et al. 2020].

3.1.1 Theory of non-Euclidean plates. Let us consider a deformation from an initially flat plate of thickness h to a curved surface, defined by the mapping $\Phi : \mathcal{V} = \hat{\mathcal{S}} \times [-\frac{h}{2}, \frac{h}{2}] \rightarrow \mathbb{R}^3$. We define the curved surface as $\mathcal{S} = \phi(\hat{\mathcal{S}})$ with ϕ being a parameterization of the mid-surface (see Figure 4).

The metric $\mathbf{g} = \mathrm{d}\Phi^T \mathrm{d}\Phi$ measures the deformations in lengths and angles between the initial flat domain and the deformed domain. The elastic energy of the plate depends on its rest state and the corresponding metric $\bar{\mathbf{g}}$ measuring the deformation between the initial and rest configurations. Usually, the rest state and the initial state coincide, making $\bar{\mathbf{g}}$ the identity. However, plastic deformations like the shrinkage of PLA under heat modify the rest shape. The resulting rest state is often not embeddable in $\mathbb{R}^3$: there is likely no shape in $\mathbb{R}^3$ that would not have residual stresses. However, even

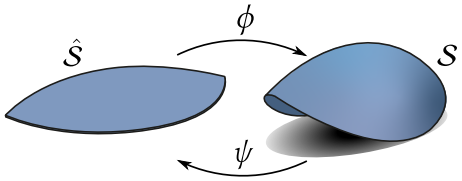


Fig. 4. $\hat{\mathcal{S}}$ represents the midsurface of the plate in the initial parametric space, $\mathcal{S}$ is the curved deformed midsurface. ϕ and $\psi = \phi^{-1}$ are the mapping between them.

though the rest shape cannot exist in $\mathbb{R}^3$ it remains possible to define $\bar{\mathbf{g}}$ for our case of plastic deformation, as detailed in Section 3.1.2.

Using standard plane-stress assumptions, we find that the metrics $\mathbf{g}$ and $\bar{\mathbf{g}}$ can be reduced to their planar 2×2 sub-parts $\mathbf{C}$ and $\bar{\mathbf{C}}$ corresponding to the deformation of the mid-surface. Assuming the thickness h is negligible compared to the dimensions of the surface $\hat{\mathcal{S}}$, $\mathbf{C}$ and $\bar{\mathbf{C}}$ can be expanded into first and second fundamental forms:

$$\mathbf{C} = \mathbf{a} - 2z\mathbf{b} + O(z^2), \;\; \bar{\mathbf{C}} = \bar{\mathbf{a}} - 2z\bar{\mathbf{b}} + O(z^2), \tag{1}$$

Integrating the total elastic energy $W = \int_{\mathcal{V}} \left\| \frac{1}{2}(\mathbf{C} - \bar{\mathbf{C}}) \right\|_{\mathrm{M}}^2 \mathrm{d}V$ in the thickness direction gives the following expansion:

$$W = \frac{h}{4} \int_{\hat{\mathcal{S}}} \|\mathbf{a} - \bar{\mathbf{a}}\|_{\mathrm{M}}^2 \, \mathrm{d}A + \frac{h^3}{12} \int_{\hat{\mathcal{S}}} \left\|\mathbf{b} - \bar{\mathbf{b}}\right\|_{\mathrm{M}}^2 \mathrm{d}A + O(h^4). \tag{2}$$

As the plate is non-Euclidean, there is not necessarily a shape in $\mathbb{R}^3$ such that $W = 0$. In the equation above, $\|\cdot\|_{\mathrm{M}}$ is a material norm. Previous methods [Chen et al. 2018; Weischedel 2012] employ a

so-called *isotropic Saint-Venant–Kirchhoff norm*, suitable for large displacements but small strains. However, since in our case the material exhibits an *orthotropic* response, we generalize this StVK norm so as to model orthotropic materials with Young's moduli E_1 in the orientation θ and E_2 in the orthogonal direction. The material norm of a matrix $\boldsymbol{\epsilon}$ is then written as:

$$\|\boldsymbol{\epsilon}\|_{\mathsf{M}} = (R_\theta^T \bar{\mathbf{a}}^{-1} \boldsymbol{\epsilon} R_\theta) : C : (R_\theta^T \bar{\mathbf{a}}^{-1} \boldsymbol{\epsilon} R_\theta) \tag{3}$$

where C is the rank-4 linear elastic tensor of the orthotropic material. We use the parameterization of Li and Barbic [2014] who write this tensor as a 3×3 symmetric matrix using Voigt notation:

$$C = \frac{1}{1-\nu^2} \begin{pmatrix} E_1 & \nu\sqrt{E_1 E_2} & 0 \\ \nu\sqrt{E_1 E_2} & E_2 & 0 \\ 0 & 0 & \frac{1-\nu}{2}\sqrt{E_1 E_2} \end{pmatrix}, \tag{4}$$

with a Poisson's ratio-like parameter ν which we set to 0.4 as it is a reasonable value for PLA. We experimentally determine an approximate ratio $E_1/E_2 = 10$ for the material we work with (see Section 4.3 for a discussion).

To find the forms $\mathbf{a}$ and $\mathbf{b}$, the deformation map Φ is computed from the parameterization of its mid-surface ϕ and its normal map n:

$$\Phi(x, y, z) = \phi(x, y) + z\rho n(x, y).$$

The growth factor ρ accounts for the shell being allowed to scale in the normal direction. Most previous work on non-Euclidean surfaces [Chen et al. 2018; Efrati et al. 2009] assumes that the thickness of a plate stays constant during the deformation, making $\rho = 1$. However, this is not the case for the shape memory effect of PLA. We instead follow Pezzulla et al. [2017] and consider that the material has a constant growth factor $\rho = \lambda_3$ along this direction, which we measure in Section 4. The thin-shell assumption therefore gives:

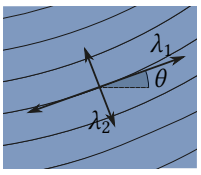
$$\mathbf{C} = \mathrm{d}\phi^T \mathrm{d}\phi + 2\rho z \mathrm{d}n^T \mathrm{d}\phi + O(z^2) = \mathbf{a} - 2z\mathbf{b} + O(z^2). \tag{5}$$

This equation allows to identify $\mathbf{a} = \mathrm{d}\phi^T \mathrm{d}\phi$ and $\mathbf{b} = -\rho \mathrm{d}n^T \mathrm{d}\phi$, with $\mathrm{d}\phi$ being the deformation gradient and $\mathrm{d}n$ is the so-called *shape operator* (see e.g. [Grinspun et al. 2006]).

Note that the integration in Equation 2 is done over the initial shape. The fundamental forms $\mathbf{a}$ and $\mathbf{b}$ are computed on $\hat{\mathcal{S}}$ as pullbacks of the fundamental forms used in, *e.g.*, [Chen et al. 2018]. We choose to use these pullbacks and define everything on $\hat{\mathcal{S}}$ as it makes the rest metric easier to define in this setting.

3.1.2 Simulation of shape memory PLA. As explained earlier, the printed PLA shrinks in the direction of the trajectories when heated. A secondary effect is that it also grows in the other directions. In the following paragraph, we estimate the rest fundamental forms $\bar{\mathbf{a}}$ and $\bar{\mathbf{b}}$ by measuring this deformation.

We call λ_1 and λ_2 the ratios measuring the in-plane distortion (*i.e.* the change of length) respectively in the path direction (represented by an angle θ) and in the orthogonal direction, and λ_3 the ratio measuring the change in length along the thickness in the normal direction. We explain in Section 4.1 how to measure experimentally these ratios, which are constant through the plate.



Noting that a vector oriented along θ will change its length by a factor λ_1 and a vector in the perpendicular direction will change its length by a factor λ_2, we write:

$$\bar{\mathbf{C}} = R_\theta \begin{pmatrix} \lambda_1^2 & 0 \\ 0 & \lambda_2^2 \end{pmatrix} R_\theta^T. \tag{6}$$

Assuming that θ is a linear function of z, we can write it as $\theta(x, y, z) = \theta_1(x, y) + \frac{z}{h}\theta_2(x, y)$ where θ_1 is the mean angle value and θ_2 represents the variation along the thickness, and make the following expansion

$$\bar{\mathbf{C}} = R_{\theta_1} \begin{pmatrix} \lambda_1^2 & 0 \\ 0 & \lambda_2^2 \end{pmatrix} R_{\theta_1}^T - \frac{z}{h}\theta_2(\lambda_2^2 - \lambda_1^2) R_{\theta_1} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} R_{\theta_1}^T + O(z^2), \tag{7}$$

which allows us to identify $\bar{\mathbf{a}}$ and $\bar{\mathbf{b}}$.

3.1.3 Discretization. We simulate the deformation induced by the shape memory effect by minimizing the energy W defined in Equation 2. We represent the mid-surface $\mathcal{S}$ as a triangular mesh composed of a set of vertices V and a set of triangles T. We consider the fields θ_1, θ_2 to be constant over each triangle. A vector of 2D coordinates describes the position of the vertices $\hat{\mathbf{x}}$ in the initial flat position and by the 3D coordinates $\mathbf{x}$ in a deformed state. We use piecewise linear finite elements so that the strain is constant on each triangle. We can then obtain the following discretized energy:

$$W(\mathbf{x}) = \left(\frac{h}{4} \sum_{t \in T} \|\mathbf{a} - \bar{\mathbf{a}}\|_{\mathsf{M}}^2 + \frac{h^3}{12} \sum_{t \in T} \left\|\mathbf{b} - \bar{\mathbf{b}}\right\|_{\mathsf{M}}^2 \right) \sqrt{\det \bar{\mathbf{a}}}. \tag{8}$$

$\bar{\mathbf{a}}$ and $\bar{\mathbf{b}}$ are computed using Equation 7 using the value of θ_1 and θ_2 of the triangle t. $\mathbf{a}$ and $\mathbf{b}$ are derived from Equation 5 as $\mathbf{a} = F^T F$ and $\mathbf{b} = F^T \Lambda F$. F is the deformation gradient obtained by the initial positions $\{\hat{v}_i\}_{1,2,3}$ of the vertices of triangle t and their deformed positions $\{v_i\}_{1,2,3}$:

$$F = \left(v_1 - v_0, v_2 - v_0\right) \left(\hat{v}_1 - \hat{v}_0, \hat{v}_2 - \hat{v}_0\right)^{-1}, \tag{9}$$

and Λ is the shape operator defined on the deformed surface. We use the triangle-averaged shape operator from Grinspun et al. [2006]:

$$\Lambda = \sum_{i=1,2,3} \frac{\theta_i}{A \ell_i} \tau_i \tau_i^T, \tag{10}$$

with A being the area of the triangle, θ_i the signed dihedral angle between t and its i^{th} neighbor, ℓ_i the length of the common edge e_i and τ_i a vector obtained by rotating e_i of 90° around the normal of the triangle.

The final shape induced by the shrinking of the plate can then be computed as $\mathbf{x}_f = \arg\min_{\mathbf{x}} W(\mathbf{x})$. We minimize W using a projected Newton strategy, computing derivatives and projecting the per-triangle Hessians on the PSD cone using TinyAD [Schmidt et al. 2022].

3.2 Inverse problem

The forward problem lets us predict the curved surface obtained by heating a flat printed object. We now tackle the inverse problem: given a curved surface $\mathcal{S}$ representing the mid-surface of the plate, we want to find a flat shape and the printing trajectories such that it will deform into $\mathcal{S}$ when heated above glass temperature.

Namely, we are looking for a parametric domain $\hat{\mathcal{S}}$ and the θ_1 and θ_2 fields defined on it such that, when applying the forward simulation, they deform $\hat{\mathcal{S}}$ into $\mathcal{S}$.

We decompose this problem in two parts. First, we consider that the direction field is constant across the thickness, setting $\theta_2 = 0$. This lets us optimize $\hat{\mathcal{S}}$ and θ_1, only considering the in-plane deformations. Second, to accurately control the bending behavior of the surface, we optimize for orientation variations in the thickness θ_2. We use sensitivity analysis [Zehnder et al. 2021], minimizing the distance between the simulated and target shapes (Section 3.2.2). Finally, from the field $\theta = \theta_1 + \frac{z}{h}\theta_2$, we compute the printing trajectories for each plate layer.

3.2.1 Parameterization. As a first step we find a mapping $\psi : \mathcal{S} \to \mathbb{R}^2$ whose induced distortions correspond to the shape memory effect of the PLA as close as possible. So the differential $\mathrm{d}\psi \in \mathbb{R}^{2\times 2}$ – parameterized by local barycentric coordinates on tangent planes – has singular values equal to λ_1^{-1} and λ_2^{-1}. By optimizing ψ we also obtain the parameterization domain $\hat{\mathcal{S}} = \psi(\mathcal{S})$ as well as the average orientation θ_1 — the orientation on the mid-surface — as the direction of the maximal distortion of ψ.

We follow the local/global optimization scheme of Aharoni et al. [2018]. The energy to minimize is:

$$E(\psi, U, V) = \int_{\mathcal{S}} \left\| \mathrm{d}\psi - U \begin{pmatrix} \lambda_1^{-1} & 0 \\ 0 & \lambda_2^{-1} \end{pmatrix} V^T \right\|_F^2 \mathrm{d}A. \tag{11}$$

The idea is to alternate between minimizing for ψ while keeping U, V constant on the one hand (the "global" problem) and minimizing U and V while treating $\mathrm{d}\psi$ as a constant on the other (the "local" problem). We follow the framework of Liu et al. [2008] that we briefly remind below.

The global problem is solved by fixing U and V (from the initialization or a previous iteration) and minimizing for ψ, it is solved by solving a linear system of $2n_V$ linear equations $-n_V$ being the number of vertices– whose unknowns are the discretized positions $\hat{\mathbf{x}}$ representing $\hat{\mathcal{S}}$ in $\mathbb{R}^2$.

Then, by computing the singular value decomposition (SVD) of $\mathrm{d}\psi = R_\alpha \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} R^T$ where R_α is a rotation of angle α such that $u_\alpha = (\cos\alpha, \sin\alpha)$ is the direction of maximum shrink, the local problem admits the closed-form solution $U = R_\alpha$, $V = R$.

Iterating between these two problems quickly converges to a discrete mapping ψ matching the prescribed distortions in a least squares manner. Moreover we also get the associated flat domain $\hat{\mathcal{S}} = \psi(\mathcal{S})$ describing the flat plate to be printed and we can deduce the directions θ_1 from the SVD of $\mathrm{d}\psi$ since u_α represents the direction of maximum shrinkage.

The local-global algorithm needs an initial parameterization. Unfortunately, different initializations can push the optimization to converge to vastly different local minima, some of which are not suitable for our fabrication purposes. In particular, we need a distribution of directions that are smooth enough to be able to trace continuous deposition trajectories. We found that first running as-rigid-as-possible (ARAP) parameterization (i.e., running the same algorithm with $\lambda_1 = \lambda_2 = 1$) and then deforming the flat surface by a factor λ_1^{-1} along one axis and λ_2^{-1} provides a good initialization to this algorithm, which encourages smooth directions u_α in solutions. In practice, this worked well across all our examples.

In many cases, this initialization was enough to ensure smooth directions. If extra smoothness is required it is also possible to add to the optimization problem an explicit smoothness term as in Panetta et al. [2021]. In their case, the smoothness term consists of the Dirichlet energy of the vector field of distortion directions $u_{2\alpha}$ (where α is doubled to account for the π symmetry of the 2-RoSy direction field):

$$\min_{\psi} \int_{\mathcal{S}} \left\| \begin{pmatrix} \sigma_1 - \lambda_1^{-1} & 0 \\ 0 & \sigma_2 - \lambda_2^{-1} \end{pmatrix} \right\|_F^2 \mathrm{d}A + w_D \int_{\mathcal{S}} \|\nabla u_{2\alpha}\|^2 \, \mathrm{d}A. \tag{12}$$

Since there is no obvious way to solve for this in the local-global framework, we propose to solve Equation 12 using a projected Newton strategy and computing the derivatives using TinyAD [Schmidt et al. 2022]. The difference between running this smoothing step or not is illustrated in Figure 5. As this smoothing step may not always be necessary, we show the cases where we use it in Table 1 as $w_D = 0.1$.

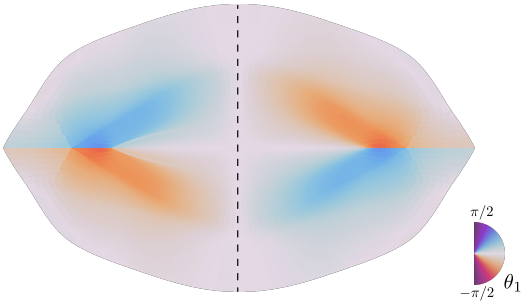


Fig. 5. Comparison of the parameterization with and without the additional smoothness term. *Left:* result of the local/global algorithm without any smoothness enforced, *right:* result with the additional smoothing step. The maximum shrinkage angle $\alpha = \theta_1$ is displayed here in the range $[-\pi/2, \pi/2]$.

3.2.2 Directions optimization. The previous optimization gives us both a domain $\hat{\mathcal{S}}$ and average directions θ_1 minimizing $\|\mathbf{a} - \bar{\mathbf{a}}\|_{\mathsf{M}}^2$. We now seek to optimize orientation variations across the thickness θ_2. This parameter controls the shell mean curvature / bending. When optimizing for θ_2 we keep θ_1 and $\hat{\mathcal{S}}$ fixed.

This problem cannot be solved by simply finding the fundamental forms $\bar{\mathbf{a}}$ and $\bar{\mathbf{b}}$ which minimize Equation 2, i.e., directly minimizing $\|\mathbf{b} - \bar{\mathbf{b}}\|_{\mathsf{M}}^2$. Indeed, contrary to the $\bar{\mathbf{a}}$ case we cannot, in general, find θ_2 such that $\|\mathbf{b} - \bar{\mathbf{b}}\|_{\mathsf{M}} \approx 0$. This means that even if we minimize the energy W as a function of θ_1 and θ_2 in Equation 2, the target shape will not correspond to a stable equilibrium, and the simulation might significantly drift from the target shape.

We propose instead to minimize the distance between the target shape $\mathcal{S}$ and the result of the forward simulation applied on θ_2 with $\theta_1, \hat{\mathcal{S}}$ fixed. We formulate the problem directly in the discretized domain. We consider vectors containing the position of all the vertices of the mesh. We call $\mathbf{x}_T$ the positions of the target shape, while $\mathbf{x}$ are the positions resulting from a forward simulation. We also use a

vector $\boldsymbol{\theta_2}$ containing the values of θ_2 per vertex. The values of θ_2 are averaged per triangle for the simulation.

The objective is to find the value of $\boldsymbol{\theta_2}$ that minimizes the distance between $\mathbf{x}_T$ and $\mathbf{x}$. The forward simulation results in a shape at equilibrium, giving us the constraint $\frac{\partial W}{\partial x} = 0$.

We add two regularization terms to the minimization problem. First, ${\boldsymbol{\theta_2}}^T M \boldsymbol{\theta_2}$ (where M is the vertex lumped mass matrix) ensures that θ_2 is on the order of h (and therefore small). Second, ${\boldsymbol{\theta_2}}^T L \boldsymbol{\theta_2}$ where L is the cotan Laplacian. This ensures that the resulting field θ remains integrable. We obtain the following constrained optimization problem:

$$\min_{\boldsymbol{\theta_2}} (\mathbf{x} - \mathbf{x}_T)^T M (\mathbf{x} - \mathbf{x}_T) + w_M {\boldsymbol{\theta_2}}^T M \boldsymbol{\theta_2} + w_L {\boldsymbol{\theta_2}}^T L \boldsymbol{\theta_2} \quad \text{subject to } \frac{\partial W}{\partial x}(\mathbf{x}, \boldsymbol{\theta_2}) = 0. \tag{13}$$

This problem can be converted to an unconstrained, nonlinear least squares program by expressing $\mathbf{x}$ as an implicit function of $\boldsymbol{\theta_2}$ through the $\frac{\partial W}{\partial x} = 0$ constraint. $\mathbf{x}(\boldsymbol{\theta_2})$ can be evaluated by solving the forward problem (Section 3.1). The energy to be minimized can then be expressed as a function of $\boldsymbol{\theta_2}$: $E(\boldsymbol{\theta_2}) = (\mathbf{x}(\boldsymbol{\theta_2}) - \mathbf{x}_T)^T M (\mathbf{x}(\boldsymbol{\theta_2}) - \mathbf{x}_T) + w_M {\boldsymbol{\theta_2}}^T M \boldsymbol{\theta_2} + w_L {\boldsymbol{\theta_2}}^T L \boldsymbol{\theta_2}$.

$E(\boldsymbol{\theta_2})$ can then be differentiated using the adjoint method from sensitivity analysis as done by Zehnder et al. [2021]:

$$\frac{\partial E}{\partial \boldsymbol{\theta_2}} = -2 \frac{\partial^2 W}{\partial \boldsymbol{\theta_2} \partial \mathbf{x}} \left(\frac{\partial^2 W}{\partial \mathbf{x}^2} \right)^{-1} M(\mathbf{x} - \mathbf{x}_T) + 2 w_M M \boldsymbol{\theta_2} + 2 w_L L \boldsymbol{\theta_2}. \tag{14}$$

We then minimize this nonlinear least squares optimization problem using the method of Zehnder et al. [2021]. Optimizing the thickness-wise variations θ_2 allows a significant decrease in the deviation between the target surface and the simulation (see Figure 6). Note that even though the difference between layer trajectories is small (see Figure 7), the induced deformation can be large.

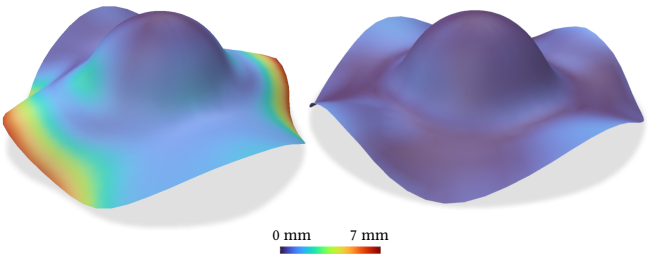


Fig. 6. Comparison with and without the optimization of θ_2. Distance ranges between 0 and 7 mm (for a 50mm-wide object).

3.2.3 Trajectory generation. The final step is to extract oriented trajectories covering the plate $\hat{\mathcal{S}}$ while following the field θ in each layer. The material will be deposited along these trajectories during fabrication. We fabricated the plate with n different layers, so that each layer has orientations $\theta^k = \theta_1 + (\frac{k-1}{n-1} - \frac{1}{2})\theta_2, \ k \in [1, n]$. We then use the Stripe Patterns algorithm [Knöppel et al. 2013] to extract a set of equally spaced curves covering the domain and following the input direction field. This algorithm takes as input a 2-rotationally symmetric (2-RoSy) direction field in complex form, which in our case amounts to writing $e^{2i\theta^k}$.

Finally, we order the trajectories by solving a Traveling Salesperson Problem (TSP), minimizing the traveling distance between curve endpoints. We use ORTools [Perron and Furnon 2023] using the Guided Local Search metaheuristic initialized by the Savings algorithm. As this NP-hard problem may take a long time to find a global minimum, we set a time limit of 5 seconds for each layer.

This set of ordered trajectories (displayed in Figure 3: Trajectory generation, with colors indicating the ordering from white to blue) can then be converted into instructions for the printer – typically G-Code – that is used to fabricate the plate.

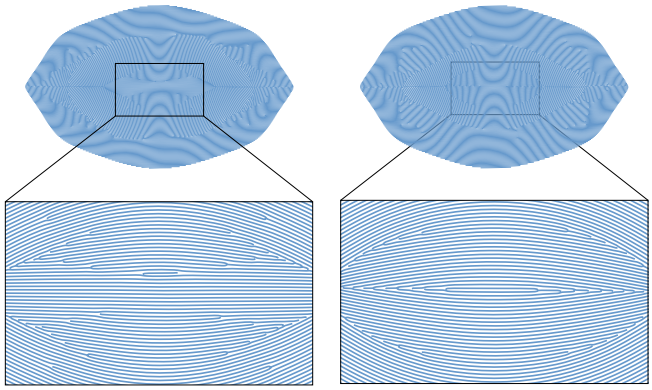


Fig. 7. Comparison between the first (*left*) and last (*right*) generated layers, with closeups on the bottom row.

4 MATERIALS STUDY

We discuss in this Section our experiments in finding the material parameters to feed to the inverse design optimization.

4.1 Measurements for $\lambda_1, \lambda_2, \lambda_3$

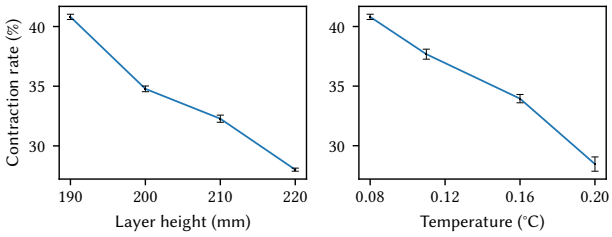


Fig. 8. Experiments with varying the extrusion temperature and layer height: lower temperatures and smaller layer heights result in higher contraction rates.

To find the distortion ratios λ_1, λ_2 and λ_3, we print 80 mm × 20 mm × 0.8 mm cuboids with parallel printing trajectories aligned with the longest sample dimension (see Figure 9). We heat each

sample by immersing them in a 90°C water bath and measure their final dimensions.[1]

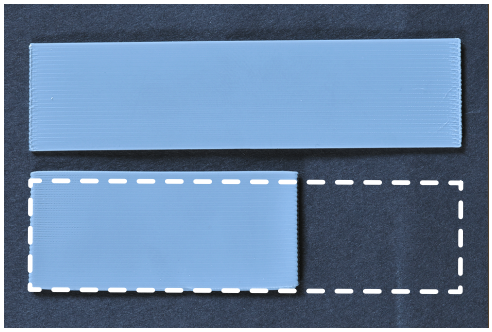


Fig. 9. A sample before (top) and after shrinkage (bottom).

We measured the distortion ratios $(\lambda_1, \lambda_2, \lambda_3)$ for several sets of printing parameters with varying extrusion temperatures and layer heights. For each set, we printed 4 samples, measured their lengths before and after shrinking, respectively l_0 and l and computed the distortion ratio $\lambda = \frac{l}{l_0}$. We can equivalently (as was done in previous work) compute a contraction rate $c = 100\frac{l-l_0}{l_0}$ (in %, see Figure 8). Despite small variations, we observed that the measurements of $\lambda_1, \lambda_2, \lambda_3$ were consistent across each set, standard deviations are displayed as error bars in Figure 8. We choose to use the average of a set for each ratio.

During our experiments, we noticed that the shrinking behavior depends heavily on the PLA used, with filaments from different vendors and colors showing varying shrinkage performance. We believe these differences are due to additives used in the composition of the PLA-based filament. For a given choice of filament, we observed that factors such as layer height, deposition widths, and nozzle temperature impact the shrinkage magnitude. We, however, found no dependency on the printing speeds contrary to prior works [Kačergis et al. 2019; Rajkumar and Shanmugam 2018]. This, again, may be due to differences in exact filament formulations.

For the material we used in our examples –Polymaker Polymax PLA (grey)– the best contraction rate we obtain is by using an extrusion temperature of 190°C with a layer height of 0.08mm and a nozzle of 0.4mm diameter. We obtain a contraction rate of 41% which is higher than previous work on heat-shrinkable PLA [Gu et al. 2019; Rajkumar and Shanmugam 2018; van Manen et al. 2017], even though better performance can be obtained with more exotic materials (see e.g. [Chalissery et al. 2022]).

We use these parameters in our examples and the corresponding ratios $\lambda_1 = 0.58$, $\lambda_2 = 1.08$, and $\lambda_3 = 1.52$.

4.2 Taking the non-zero rest curvature into account

Besides shrinking and expanding of ratios $\lambda_1, \lambda_2, \lambda_3$, the test cuboid samples also exhibited a constant, non-zero curvature after being heated.

[1]Note that the presence of water is not necessary for the shape memory effect to occur but merely provides a suitable environment in which the temperature stays stable.

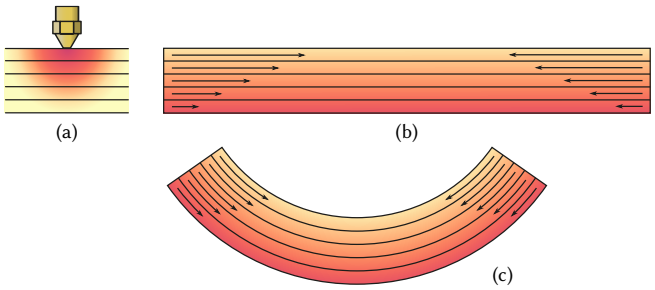


Fig. 10. *Bilayer effect.* When the nozzle extrudes material for a top layer, the layers below are re-heated locally *(a)*. As a consequence, the bottom layers accumulate more heat than the top ones. We hypothesize that this relaxes the stresses within the bottom layers more than for the top layers *(b)*. This translates to a linear variation of shrinkage which induces curvature in the final sample *(c)*.

We hypothesize that this is due to re-heating during the layer-by-layer fabrication. As illustrated in Figure 10, the lower layers are partially annealed by the hot nozzle re-heating them when depositing material above. This diminishes their shrinkage performance and, in turn, causes a bilayer effect: the bottom layers shrink less than the top layers, biasing the surface to curve downwards.

We, therefore, incorporated this bilayer effect into the material model, considering that λ_1 varies linearly across the thickness. We modify the rest metric Equation 6 as follows

$$\bar{\mathbf{C}} = R_\theta \begin{pmatrix} (\lambda_1 + \frac{z}{h}\Delta\lambda_1)^2 & 0 \\ 0 & \lambda_2^2 \end{pmatrix} R_\theta^T \tag{15}$$

where $\Delta\lambda_1$ measures the variation of λ_1 according to z. We assume here that $\Delta\lambda_2 = \Delta\lambda_3 = 0$, even though λ_2 and λ_3 may also be subject to variations due to reheating during printing, we found their variations to be negligible.

We measure the natural curvature κ taken by rectangular samples using the Kappa plugin from Fiji [Schindelin et al. 2012] (Figure 11) and deduce the value of $\Delta\lambda_1$ from these measurements. The samples have parallel horizontal trajectories, so for them, we have $\theta_1 = \theta_2 = 0$, thus

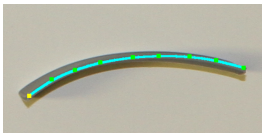


Fig. 11. Image-based measurement of the curvature κ.

$$\bar{\mathbf{C}} = R_\theta \begin{pmatrix} \lambda_1^2 & 0 \\ 0 & \lambda_2^2 \end{pmatrix} R_\theta^T + 2\frac{z}{h}\Delta\lambda_1\,\lambda_1 R_\theta \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} R_\theta^T + O(z^2). \tag{16}$$

We identify the second fundamental form as

$$\bar{\mathbf{b}} = -\frac{\Delta\lambda_1\lambda_1}{h} R_\theta \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} R_\theta^T. \tag{17}$$

Making the simplifying assumption that the samples form ribbons with constant curvature κ when deforming, we find the second fundamental form is constant on the surface

$$\mathbf{b} = \begin{pmatrix} \lambda_3\lambda_1^2\kappa & 0 \\ 0 & 0 \end{pmatrix}, \tag{18}$$

therefore $\mathbf{b} = \bar{\mathbf{b}}$ gives us $\Delta\lambda_1 = -h\lambda_3\lambda_1\kappa$.

We only need to modify the forward method using Equation 15 to take this effect into account. The modification is naturally integrated into the inverse problem through the constraint $\frac{\partial W}{\partial x} = 0$ that uses the forward simulation. Note that other phenomena that could occur for different materials may be added to the method similarly by only correcting the forward simulation.

4.3 Estimating the directional moduli

As described in Section 3.1.1, we observed that our material has an orthotropic behavior. We could however not measure the exact directional Young's moduli of the material. Doing proper measurements to capture the value of E_1 and E_2 is indeed challenging: one would have to set up cantilever tests inside a thermal chamber at 90°C and then find a way to recover these coefficients from the experimental results, which is far from trivial since the test samples have non-zero rest curvature. Therefore we leave such precise measurements as future work.

Instead we experimentally set the ratio E_1/E_2 such that the simulation matches a test example printed for this purpose (Figure 12). The example, initially shaped like an ellipse squished on the poles, deploys into what looks like a pinched hemisphere because of the orthotropic behavior of the material (Figure 12 right): it is easier to bend across the trajectories than along them. The simulation using $E_1 = E_2 = 1$ gives a shape with a near constant curvature that does not match the fabricated object (Figure 12 left) since it does not account for the orthotropy. We obtain a much closer simulation result using $E_1 = 10$ and $E_2 = 1$ (Figure 12 middle).

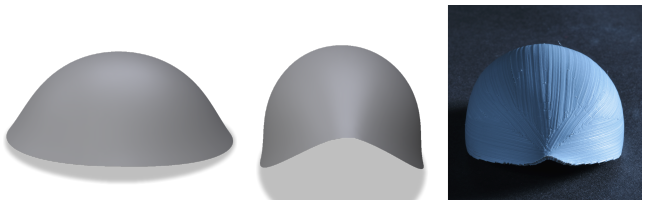


Fig. 12. Using the same initial state, we compare the simulation with $E_1/E_2 = 1$ (*left*), the simulation with $E_1/E_2 = 10$ (*middle*), and the fabricated example (*right*).

5 RESULTS

We demonstrate our method on a variety of target surfaces, visible in Figures 1, 3 and 13. We printed our examples on a Creality CR10S Pro using Polymaker Polymax PLA (grey) and immersed them in a 90°C water contained heated using an Anova Precision sous-vide cooker. The simulation and inverse problems are computed on a 6-core Intel Xeon E-2186G processor.

The first two rows of Figure 13 show a hemisphere and a saddle having positive and negative Gaussian curvature, respectively. The Beetle example in Figure 1 reproduces the hood of a VW Beetle, which, together with the Hat and Mask model in Figure 13 showcase more complex surfaces with spatially varying curvatures. The shells we create are typical of covers designed for enclosures and appliances (e.g., a computer mouse, a hand rest, a game console, a curved phone stand) and would be very difficult to fabricate otherwise. As they start as flat plates, they all have a smooth surface finish. They are easily flexed, showing no fragility in the bending direction.

Please refer to the accompanying video for heating and deformation sequences and turntable inspection of the results.

We provide optimization performance timings and printing times for these examples in Table 1. The optimization typically takes less than one hour. The field optimization is usually the most expensive step, followed by trajectory extraction. Printing times range from one to four hours, depending on the plate size. Note that both processes require no user intervention (apart from removing the fabricated plate from the printer): the optimization runs autonomously, and the printing is made trivial by the fact that we fabricate a simple plate. Both could be optimized further to reduce the time taken.

The "Dist. to sim." and "Dist. to optim." columns in Table 1 measure the distance between the simulated model before/after optimizing θ_2 and the target shape. As can be seen, the optimized surface is a close fit to the target with only a few percents max deviation.

Of course, the fabrication process introduces numerous approximations. First and foremost, the discretization in trajectories and layers, followed by imprecisions in the printer setup and uncertainties in the material parameters. Thus, while the results are visually similar to the target shapes, we measured the actual error between the printed result and the target shape for the beetle example. We scanned and reconstructed the fabricated shell (visible in Figure 1) using the Meshroom photogrammetry software [Griwodz et al. 2021]. We then measured the distance between each point of the reconstructed model and the target mesh. The error is visualized in Figure 14, as well as the superposition between the target and reconstructed surfaces. We obtain an average error distance of 2.2mm and a maximal distance of 9.7mm for a model of size 129mm. While clearly imperfect, it would be impossible to design such a plate by hand that would come even remotely close to the target. For cases where a tight fit to another object is required, the natural flexibility and strength of the shell would easily allow to snap-fit it in place.

Finally, Figure 6 demonstrates the importance of optimizing the trajectory orientations across the plate thickness (θ_2), on the left the example was simulated with $\theta_2 = 0$ while the example on the right is the output of the trajectory optimization of Section 3.2.2.

6 LIMITATIONS AND FUTURE WORK

Developable surfaces. Our approach sometimes fails at reproducing developable surfaces. For example, we printed a near-developable fashion surface model with one piece wrapped around the body. Even though the optimization succeeded at satisfying the constraints (see how the deviations were greatly minimized, model *Top* in Table 1), the plate did not wrap around the correct geometry, as shown in Figure 15. In such a case, the user may manually guide the plate towards the target when it is still hot, as shown in Figure 15, right.[2]

This issue appears as developable surfaces are a very specific case for our approach: they undergo no change in length when deforming, while our choice of optimization strategy is based on changes in length. There is also no secondary curvature to guide

[2]We did not guide any other result into shape, see accompanying video for the dipping process

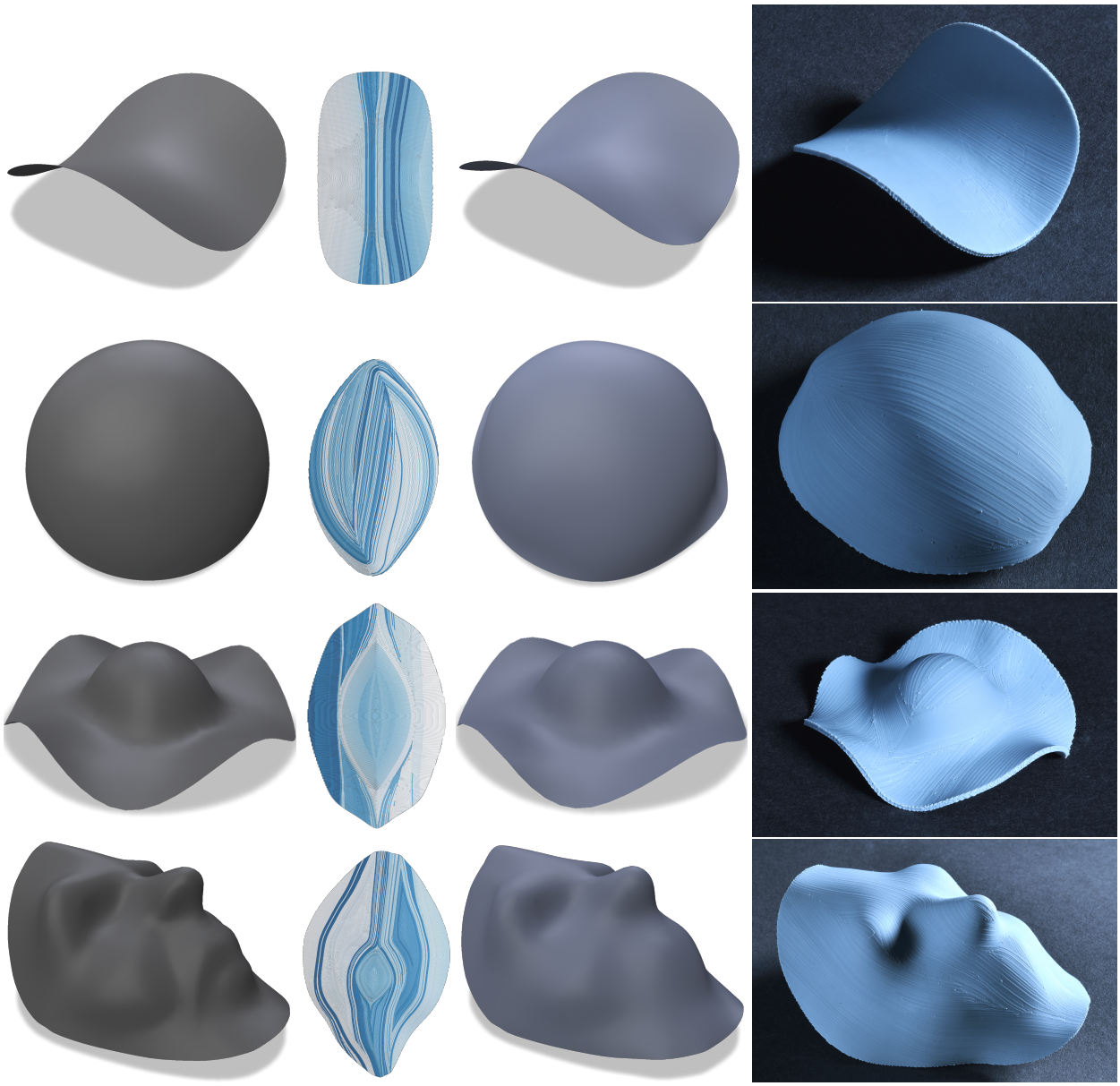


Fig. 13. Our results for the models saddle, hemisphere, hat and face: (*from left to right*) the target mesh, our optimized printing trajectories, the resulting simulated shape and the fabricated object.

the deformation and rigidify the shell. As a result, our optimizer does not perform well in these cases.

Nevertheless, since θ_2 allows us to control normal curvature, it is possible to find θ_1, θ_2 that would allow to create a developable surface such as a cylinder for example. However, our optimization strategy is not well suited for this case and does not always yield the best value for θ_1. How to robustly achieve this within a more general optimization framework is a clear path for future work.

High frequency details. The thickness of the plates imposes a natural limit to the achievable normal curvature, which in turn limits the achievable details. This is properly captured in simulation (see the face model in Figure 13) such that its impact on the result is predictable. Scaling up the design for a same thickness can help capture more high-frequency details relative to the scale.

Multistability in shells. A shell structure can exhibit multiple stable configurations but typically tends to deform towards the configuration with the lowest energy. As we run a static simulation at each iteration of our optimization, the lowest-energy state should be the one closest to the input shape – it was the case in all our results.

Table 1. For each result shown in the paper, we report the computation times separated between parameterization $\mathbf{t_{param}}$, directions optimization $\mathbf{t_{optim}}$ and trajectory generation $\mathbf{t_{traj}}$, the print time, the distance between the simulated shape and the target before/after optimizing θ_2 (given as percentages of the bounding box diagonal), the distance between the optimized shape and the target, the Dirichlet smoothing parameter $w_D = 0.1$, and the height of the initial plate in mm.

Model	**Nb. faces**	$\mathbf{t_{param}}$ // $\mathbf{t_{optim}}$ // $\mathbf{t_{traj}}$ **(min'sec")**	**Print time (h:mm)**	**Dist. to sim. (avg // max)**	**Dist. to optim. (avg // max)**	$\mathbf{w_D}$	**height** (mm)
Hemisphere	17152	10.7" // 2'16" // 1'07"	2:26	3.36 // 10.28	0.25 // 3.02	0.1	150
Saddle	2048	0.1" // 0'04" // 0'58"	1:11	2.20 // 6.05	0.47 // 2.15	0	100
Beetle	8448	1.7" // 2'01" // 2'21"	2:49	5.41 // 16.23	0.34 // 1.41	0	200
Face	3996	1.1" // 0'49" // 3'50"	4:18	2.81 // 12.56	0.36 // 2.04	0.1	200
Top	3584	0.7" // 1'01" // 1'28"	2:53	5.85 // 45.62	0.24 // 2.95	0	250
Hat	11008	16.5" // 0'57" // 1'03	1:08	1.34 // 9.31	0.30 // 1.18	0.1	100

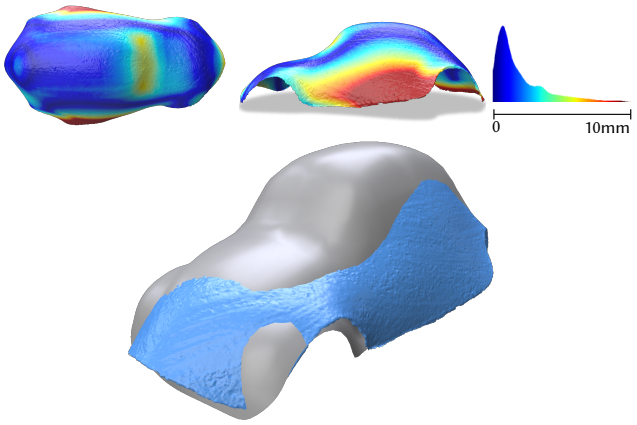


Fig. 14. Comparison between the fabricated beetle and the target mesh: (*top*) distance between fabricated and target visualized on the fabricated object, (*bottom*) superposition of the fabricated object (blue) and target surface (grey).

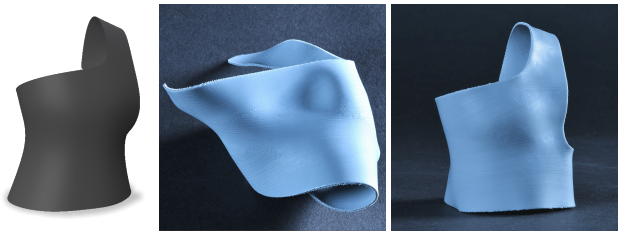


Fig. 15. A failure case: (*left*) target surface, (*middle*) deployed surface after heating, (*right*) deployed surface manually shaped when still hot.

Two-step directions optimization algorithm. The first part of our optimizer consists in finding a flat parameterization domain $\hat{\mathcal{S}}$ in which we can later optimize the directions (Section 3.2.1). A side effect of this parameterization step is we can already get an initial value for θ_1 by computing the singular vector corresponding to the maximum shrinking direction. In our experiments, we found that this initial value of θ_1 is sufficient so that we only optimize for θ_2 in the second step (Section 3.2.2), which effectively translates into a greedy algorithm with two distinct phases. Nevertheless, it could be possible to add θ_1 as optimization variables in Equation 13 and therefore consider both θ_1 and θ_2 in the optimization, but it would come at the cost of a more challenging optimization landscape and longer computation times.

Mesh quality and refinement. Both the parameterization and the simulation objectives are sensitive to mesh quality. Therefore, we make sure to use evenly-tessellated meshes and apply Loop subdivision on them until the number of vertices is around 5000, which in our experiments strikes a good balance between computation time and mesh quality.

7 CONCLUDING REMARKS

We propose a method that optimizes a plate and the orientation of its fabrication trajectories such that when dipped in hot water, it deforms into a target surface. Compared to prior methods using a similar shrinkage technique, our method is the first to rely on continuous variations in trajectory orientation across the plate and its layers.

Our technique works with commercially available PLA filament with minimal calibration on standard FFF 3D printers. Thus, we expect it to be widely adopted in the maker, hobbyist, design, and education communities, where 3D printing is already widespread.

The inverse design technique provides a tight match to the target in simulation, but there are deviations between the target shapes and fabricated shell after deformation. Still, considering the many sources of approximations and uncertainty, we hope to have demonstrated the great potential of the approach, which is already useful for producing non-trivial shell surfaces.

Fabricating a flat object presents many advantages: printing is robust, there are no supports, no layering (staircase) defects. The plate can be shipped flat and deformed when needed. The forming process, performed in hot water, is simple and fun. It would be trivial to attach a layer atop the plate before deformation, such as conductive traces or colored patterns. We believe there are many open venues for future work, both in improving the technique and finding creative applications of the fabrication method.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful feedback. We would also like to thank the following individuals: Etienne Vouga for insightful discussions about our shell material model, Emilie Yu for

modeling the Beetle, Hat, and Top test surfaces, and the developers of the open-source libraries which we used in our work: Geometry Central [Sharp et al. 2019a], Polyscope [Sharp et al. 2019b], TinyAD [Schmidt et al. 2022], and libigl [Jacobson et al. 2018]. This work was funded by the Inria Challenge "DORNELL".

REFERENCES

Hillel Aharoni, Yu Xia, Xinyue Zhang, Randall D. Kamien, and Shu Yang. 2018. Universal inverse design of surfaces with thin nematic elastomer sheets. *Proceedings of the National Academy of Sciences* 115, 28 (2018), 7206–7211. https://doi.org/10.1073/pnas.1804702115

Byoungkwon An, Ye Tao, Jianzhe Gu, Tingyu Cheng, Xiang 'Anthony' Chen, Xiaoxiao Zhang, Wei Zhao, Youngwook Do, Shigeo Takahashi, Hsiang-Yun Wu, Teng Zhang, and Lining Yao. 2018. Thermorph: Democratizing 4D Printing of Self-Folding Materials and Interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. 12 pages. https://doi.org/10.1145/3173574.3173834

Dilip Chalissery, Dennis Schönfeld, Mario Walter, Inga Shklyar, Heiko Andrae, Christoph Schwörer, Tobias Amann, Linda Weisheit, and Thorsten Pretsch. 2022. Highly Shrinkable Objects as Obtained from 4D Printing. *Macromolecular Materials and Engineering* 307, 1 (2022). https://doi.org/10.1002/mame.202100619

Hsiao-Yu Chen, Arnav Sastry, Wim M. van Rees, and Etienne Vouga. 2018. Physical Simulation of Environmentally Induced Thin Shell Deformation. *ACM Trans. Graph.* 37, 4, Article 146 (July 2018), 13 pages. https://doi.org/10.1145/3197517.3201395

Tian Chen, Julian Panetta, Max Schnaubelt, and Mark Pauly. 2021. Bistable Auxetic Surface Structures. *ACM Trans. Graph.* 40, 4, Article 39 (July 2021), 9 pages. https://doi.org/10.1145/3450626.3459940

Tiffany Cheng, Dylan Wood, Laura Kiesewetter, Eda Özdemir, Karen Antorveza, and Achim Menges. 2021. Programming material compliance and actuation: Hybrid additive fabrication of biocomposite structures for large-scale self-shaping. *Bioinspiration & Biomimetics* 16, 5 (2021).

Himani Deshpande, Clement Zheng, Courtney Starrett, Jinsil Hwaryoung Seo, and Jeeeun Kim. 2022. Fab4D: An Accessible Hybrid Approach for Programmable Shaping and Shape Changing Artifacts. In *Sixteenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '22)*. Article 70, 7 pages. https://doi.org/10.1145/3490149.3505574

Efi Efrati, Eran Sharon, and Raz Kupferman. 2009. Elastic theory of unconstrained non-Euclidean plates. *Journal of the Mechanics and Physics of Solids* 57, 4 (2009), 762–775. https://doi.org/10.1016/j.jmps.2008.12.004

A. Sydney Gladman, Elisabetta A. Matsumoto, Ralph G. Nuzzo, L. Mahadevan, and Jennifer A. Lewis. 2016. Biomimetic 4D Printing. *Nature Materials* 15 (2016), 413–419. https://doi.org/10.1038/nmat4544

Itay Griniasty, Hillel Aharoni, and Efi Efrati. 2019. Curved Geometries from Planar Director Fields: Solving the Two-Dimensional Inverse Problem. *Physical Review Letters* 123 (Sept. 2019), 5 pages. Issue 12. https://doi.org/10.1103/PhysRevLett.123.127801

Eitan Grinspun, Yotam Gingold, Jason Reisman, and Denis Zorin. 2006. Computing discrete shape operators on general meshes. 25, 3 (2006), 547–556. https://doi.org/10.1111/j.1467-8659.2006.00974.x

Carsten Griwodz, Simone Gasparini, Lilian Calvet, Pierre Gurdjos, Fabien Castan, Benoit Maujean, Gregoire De Lillo, and Yann Lanthony. 2021. AliceVision Meshroom: An open-source 3D reconstruction pipeline. In *Proceedings of the 12th ACM Multimedia Systems Conference - MMSys '21*. ACM Press. https://doi.org/10.1145/3458305.3478443

Philippe Grönquist, Dylan Wood, Mohammad M Hassani, Falk K Wittel, Achim Menges, and Markus Rüggeberg. 2019. Analysis of hygroscopic self-shaping wood at large scale for curved mass timber structures. *Science Advances* 5, 9 (2019).

Jianzhe Gu, David E. Breen, Jenny Hu, Lifeng Zhu, Ye Tao, Tyson Van de Zande, Guanyun Wang, Yongjie Jessica Zhang, and Lining Yao. 2019. Geodesy: Self-Rising 2.5D Tiles by Printing along 2D Geodesic Closed Path. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. 10 pages. https://doi.org/10.1145/3290605.3300267

Jianzhe Gu, Vidya Narayanan, Guanyun Wang, Danli Luo, Harshika Jain, Kexin Lu, Fang Qin, Sijia Wang, James McCann, and Lining Yao. 2020. Inverse Design Tool for Asymmetrical Self-Rising Surfaces with Color Texture. In *Symposium on Computational Fabrication (SCF '20)*. 12 pages. https://doi.org/10.1145/3424630.3425420

Ruslan Guseinov, Connor McMahan, Jesús Pérez, Chiara Daraio, and Bernd Bickel. 2020. Programming temporal morphing of self-actuated shells. *Nature Communications* 11, Article 237 (Dec. 2020). https://doi.org/10.1038/s41467-019-14015-2

Ruslan Guseinov, Eder Miguel, and Bernd Bickel. 2017. CurveUps: Shaping Objects from Flat Plates with Tension-Actuated Curvature. *ACM Trans. Graph.* 36, 4, Article 64 (July 2017), 12 pages. https://doi.org/10.1145/3072959.3073709

Freddie Hong, Steve Hodges, Connor Myant, and David E Boyle. 2022. Open5x: Accessible 5-Axis 3D Printing and Conformal Slicing. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems (CHI EA '22)*. Article 399, 6 pages. https://doi.org/10.1145/3491101.3519782

Alec Jacobson, Daniele Panozzo, et al. 2018. libigl: A simple C++ geometry processing library. https://libigl.github.io/.

David Jourdan, Mélina Skouras, Etienne Vouga, and Adrien Bousseau. 2022. Computational Design of Self-Actuated Surfaces by Printing Plastic Ribbons on Stretched Fabric. *Computer Graphics Forum* 41, 2 (2022), 493–506. https://doi.org/10.1111/cgf.14489

Lukas Kačergis, Rytis Mitkus, and Michael Sinapius. 2019. Influence of fused deposition modeling process parameters on the transformation of 4D printed morphing structures. *Smart Materials and Structures* 28, 10 (Sept. 2019), 105042. https://doi.org/10.1088/1361-665X/ab3d18

Jungwook Kim, James A. Hanna, Myunghwan Byun, Christian D. Santangelo, and Ryan C. Hayward. 2012. Designing Responsive Buckled Surfaces by Halftone Gel Lithography. *Science* 335, 6073 (2012), 1201–1205. https://doi.org/10.1126/science.1215309

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

Felix Knöppel, Keenan Crane, Ulrich Pinkall, and Peter Schröder. 2015. Stripe Patterns on Surfaces. *ACM Trans. Graph.* 34, Article 39 (July 2015), 10 pages. Issue 4. https://doi.org/10.1145/2767000

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 (July 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, Article 106 (July 2018), 13 pages. https://doi.org/10.1145/3197517.3201373

Yijing Li and Jernej Barbic. 2014. Stable Orthotropic Materials. In *Symposium on Computer Animation*. 41–46.

Ligang Liu, Lei Zhang, Yin Xu, Craig Gotsman, and Steven J. Gortler. 2008. A Local/Global Approach to Mesh Parameterization. In *Proceedings of the Symposium on Geometry Processing (SGP '08)*. 1495–1504. https://doi.org/10.5555/1731309.1731336

Kongpyung (Justin) Moon, Haeun Lee, Jeeeun Kim, and Andrea Bianchi. 2022. ShrinkCells: Localized and Sequential Shape-Changing Actuation of 3D-Printed Objects via Selective Heating. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22)*. Article 86, 12 pages. https://doi.org/10.1145/3526113.3545670

Jun-Hee Na, Nakul P. Bende, Jinhye Bae, Christian D. Santangelo, and Ryan C. Hayward. 2016. Grayscale gel lithography for programmed buckling of non-Euclidean hydrogel plates. *Soft Matter* 12 (2016), 4985–4990. Issue 22. https://doi.org/10.1039/C6SM00714G

Julian Panetta, Florin Isvoranu, Tian Chen, Emmanuel Siéfert, Benoît Roman, and Mark Pauly. 2021. Computational Inverse Design of Surface-Based Inflatables. *ACM Trans. Graph.* 40, 4 (July 2021), 14 pages. https://doi.org/10.1145/3450626.3459789

Jesús Pérez, Miguel A. Otaduy, and Bernhard Thomaszewski. 2017. Computational Design and Automated Fabrication of Kirchhoff-Plateau Surfaces. *ACM Trans. Graph.* 36, 4, Article 62 (July 2017), 12 pages. https://doi.org/10.1145/3072959.3073695

Laurent Perron and Vincent Furnon. 2023. *OR-Tools v9.6*. Google. https://developers.google.com/optimization/

Matteo Pezzulla, Norbert Stoop, Xin Jiang, and Douglas Holmes. 2017. Curvature-driven morphing of non-Euclidean shells. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 473, 2201 (May 2017). https://doi.org/10.1098/rspa.2017.0087

Abishera Ravichandra Rajkumar and Kumar Shanmugam. 2018. Additive manufacturing-enabled shape transformations via FFF 4D printing. *Journal of Materials Research* 33, 24 (2018), 4362–4376. https://doi.org/10.1557/jmr.2018.397

Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, et al. 2012. Fiji: an open-source platform for biological-image analysis. *Nature methods* 9, 7 (2012), 676–682.

Patrick Schmidt, Janis Born, David Bommes, Marcel Campen, and Leif Kobbelt. 2022. TinyAD: Automatic Differentiation in Geometry Processing Made Simple. *Computer Graphics Forum* 41, 5 (2022), 113–124. https://doi.org/10.1111/cgf.14607

Nicholas Sharp et al. 2019b. Polyscope. www.polyscope.run.

Nicholas Sharp, Keenan Crane, et al. 2019a. GeometryCentral: A modern C++ library of data structures and algorithms for geometry processing. https://geometry-central.net/.

Georg Sperl, Rahul Narain, and Chris Wojtan. 2020. Homogenized Yarn-Level Cloth. *ACM Trans. Graph.* 39, 4, Article 48 (Aug. 2020), 16 pages. https://doi.org/10.1145/3386569.3392412

Lingyun Sun, Yue Yang, Yu Chen, Jiaji Li, Danli Luo, Haolin Liu, Lining Yao, Ye Tao, and Guanyun Wang. 2021. ShrinCage: 4D Printing Accessories That Self-Adapt. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Article 433, 12 pages. https://doi.org/10.1145/3411764.3445220

Ye Tao, Youngwook Do, Humphrey Yang, Yi-Chin Lee, Guanyun Wang, Catherine Mondoa, Jianxun Cui, Wen Wang, and Lining Yao. 2019. Morphlour: Personalized

Flour-Based Morphing Food Induced by Dehydration or Hydration Method. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. 329–340. https://doi.org/10.1145/3332165.3347949

Ye Tao, Yi-Chin Lee, Haolin Liu, Xiaoxiao Zhang, Jianxun Cui, Catherine Mondoa, Mahnoush Babaei, Jasio Santillan, Guanyun Wang, Danli Luo, et al. 2021. Morphing pasta and beyond. *Science Advances* 7, 19 (May 2021). https://doi.org/10.1145/3332165.3347949

Teunis van Manen, Shahram Janbaz, and Amir A. Zadpoor. 2017. Programming 2D/3D shape-shifting with hobbyist 3D printers. *Materials Horizons* 4 (2017), 1064–1069. Issue 6. https://doi.org/10.1039/C7MH00269F

Guanyun Wang, Ye Tao, Ozguc Bertug Capunaman, Humphrey Yang, and Lining Yao. 2019. A-Line: 4D Printing Morphing Linear Composite Structures. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. 12 pages. https://doi.org/10.1145/3290605.3300656

Guanyun Wang, Humphrey Yang, Zeyu Yan, Nurcan Gecer Ulu, Ye Tao, Jianzhe Gu, Levent Burak Kara, and Lining Yao. 2018. 4DMesh: 4D Printing Morphing Non-Developable Mesh Surfaces. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18)*. 623–635. https://doi.org/10.1145/3242587.3242625

Wen Wang, Lining Yao, Teng Zhang, Chin-Yi Cheng, Daniel Levine, and Hiroshi Ishii. 2017. Transformative Appetite: Shape-Changing Food Transforms from 2D to 3D by Water Interaction through Cooking. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. 6123–6132. https://doi.org/10.1145/3025453.3026019

Clarisse Weischedel. 2012. *A discrete geometric view on shear-deformable shell models*. Ph. D. Dissertation. Georg August Universität Göttingen. https://doi.org/10.53846/goediss-2453

Yuxuan Yu, Haolin Liu, Kuanren Qian, Humphrey Yang, Matthew McGehee, Jianzhe Gu, Danli Luo, Lining Yao, and Yongjie Jessica Zhang. 2020. Material characterization and precise finite element analysis of fiber reinforced thermoplastic composites for 4D printing. *Computer-Aided Design* 122 (2020). https://doi.org/10.1016/j.cad.2020.102817

Jonas Zehnder, Stelian Coros, and Bernhard Thomaszewski. 2021. SGN: Sparse Gauss-Newton for Accelerated Sensitivity Analysis. *ACM Trans. Graph.* 41, 1, Article 4 (Sept. 2021), 10 pages. https://doi.org/10.1145/3470005

Quan Zhang, Dong Yan, Kai Zhang, and Gengkai Hu. 2015. Pattern Transformation of Heat-Shrinkable Polymer by Three-Dimensional (3D) Printing Technique. *Scientific Reports* 5, Article 8936 (March 2015). https://doi.org/10.1038/srep08936

Tianyu Zhang, Guoxin Fang, Yuming Huang, Neelotpal Dutta, Sylvain Lefebvre, Zekai Murat Kilic, and Charlie C. L. Wang. 2022. S^3-Slicer: A General Slicing Framework for Multi-Axis 3D Printing. *ACM Trans. Graph.* 41, 6, Article 277 (Nov. 2022), 15 pages. https://doi.org/10.1145/3550454.3555516