



Computational design of self-shaping textiles

David Jourdan

► To cite this version:

David Jourdan. Computational design of self-shaping textiles. Graphics [cs.GR]. Université Côte d'Azur, 2022. English. NNT : 2022COAZ4008 . tel-03662318v2

HAL Id: tel-03662318

<https://theses.hal.science/tel-03662318v2>

Submitted on 13 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE DE DOCTORAT

Conception numérique de textiles auto-déployables

David Jourdan

Inria, Équipe GraphDeco

Présentée en vue de l'obtention du grade de
docteur en Informatique d'Université Côte
d'Azur

Dirigée par : Adrien Bousseau

Co-encadrée par : Mélina Skouras

Soutenue le : 29 mars 2022

Devant le jury composé de :

Olivier Baverel, Professeur, École des Ponts
ParisTech et ENSA Grenoble

Paolo Cignoni, Directeur de Recherche,
CNR-ISTI

Maud Marchal, Professeure, INSA Rennes

Bernd Bickel, Professeur, IST Austria

Etienne Vouga, Professeur, The University of
Texas at Austin

Mélina Skouras, Chargée de Recherche,
Inria Grenoble Rhône-Alpes

Adrien Bousseau, Directeur de Recherche, Inria
Université Côte d'Azur

CONCEPTION NUMÉRIQUE DE TEXTILES AUTO-DÉPLOYABLES

COMPUTATIONAL DESIGN OF SELF-SHAPING TEXTILES

Jury:

Président du jury / President of the jury

Olivier Baverel, Professeur, École des Ponts ParisTech

Rapporteurs / Reviewers

Paolo Cignoni, Directeur de Recherche, CNR-ISTI

Maud Marchal, Professeure, INSA Rennes

Examineur / Examiner

Bernd Bickel, Professeur, IST Austria

Visiteur / Visitor

Etienne Vouga, Professeur, The University of Texas at Austin

Co-encadrante / Cosupervisor

Mélina Skouras, Chargée de Recherche, Inria Grenoble Rhône-Alpes

Directeur de thèse / Thesis supervisor

Adrien Bousseau, Directeur de Recherche, Inria Université Côte d'Azur

ACKNOWLEDGEMENTS

Matérialiser un roman par l'écriture est une corvée de tâcheron, aussi sombre et fastidieuse au bout de quatre cent pages qu'il était un trouble divin et une ivresse de créateur d'en imaginer, souvent durant des années, les péripéties gratuitement

— Blaise Cendrars, *L'homme foudroyé*

This thesis wouldn't have been possible without my supervisor Adrien Bousseau. Thank you for giving me the opportunity of doing all of this work, I got to spend four years learning a lot and working on simulation and fabrication and it was all because you chose to bet on me in the first place. I am also grateful for my co-advisor Mélina Skouras, even though we were not working in the same cities, you taught me so much about simulation, optimization and inverse design, especially during the few visits I got to do in Grenoble.

All the work in this thesis would not have been achieved without Etienne Vouga and Victor Romero who collaborated with us on the different research projects. Etienne, you are like a third supervisor for me, I still remember the day you decided to hop on the printing-on-fabric train with us, it was on the beginning of my PhD and you stuck with us ever since. Victor, it was fun doing experiments with you, I hope we will keep on doing cool research about interesting physical phenomena!

Thanks to Stavros, Gilles, Felix, Siddhant, Yorgos, Julien, Johanna, Valentin, Nicolas and all the other (past or present) members of the Graphdeco team I got to meet for making it such a nice environment to work in, I hope I will see you all very soon. One member of the team I want to thank in particular is Emilie Yu, not only you are an integral part of that nice Graphdeco atmosphere I just mentioned, but for me, during this PhD you were so much more. First of all, you helped us in uncountable ways for the two papers we submitted, taking nice pictures, making cool rendering or animations for illustrative purposes, you taught me how to use Blender, and you even set up the whole photogrammetry system for scanning our fabricated objects! On a more personal level, I have to say that doing a PhD is far from easy, especially during a pandemic, and I want to thank you for your emotional support during times when it was getting particularly tough on me, you are a wonderful life partner.

I am thankful for my family, and in particular for my parents who helped me financially pursuing my studies, and encouraged me to study something that I like even though it makes no sense to them.

Thank you to the members of my thesis committee, Maud Marchal, Paolo Cignoni, Bernd Bickel and Olivier Baverel for your insightful comments and very interesting questions, I wish I will get to meet you all in flesh soon. Finally, I want to thank the people and organizations who made all this work easier: Alexandra Elbakyan for making it possible to access many of the papers cited thereafter which I wouldn't have been able to read otherwise; the creators of open-source libraries that I used during my PhD, libigl [59] and polyscope [120]; and the ERC Starting Grant D3 (ERC-2016-STG 714221) which funded most of my PhD.

RÉSUMÉ

Nous décrivons une technique de fabrication récente permettant de créer des *textiles auto-déployables* qui consiste à extruder du plastique sur un tissu étiré au préalable en utilisant des procédés de fabrication additive. Cette technique d'impression sur tissu a récemment gagné en popularité car c'est une technique de fabrication peu coûteuse permettant de créer des structures *déployables* qui prennent automatiquement une forme en 3D lorsque le tissu est relâché.

Le comportement en déformation de ce matériau composite peut s'expliquer par une combinaison d'effets géométriques intrinsèques et extrinsèques, le plastique déposé étant à la fois la cause d'un changement de métrique et d'un effet bilame par la combinaison qu'il forme avec la couche sous-jacente de tissu. Nous exploitons ces observations pour créer des métamatériaux qui agissent sur ces deux effets : la métrique du tissu est modulée par la densité des motifs imprimés, et la flexion de la surface est contrôlée par l'épaisseur de plastique imprimé grâce à l'effet bilame.

Nous utilisons ces métamatériaux pour proposer deux types d'outils de design aidant à la conception de structures auto-déployables légères : d'un côté des outils de design direct, ou de recherche de forme, qui permettent de prédire la forme déployée de la structure à partir du motif à imprimer, de l'autre un outil de design inverse qui prend en entrée une forme donnée et trouve les paramètres optimaux du métamatériau (notamment l'épaisseur et la densité) pour reproduire au mieux la forme cible.

Mots-clés: structures auto-déployables – impression 3D – métamatériaux – bilame

ABSTRACT

We describe a recent manufacturing technique to design *self-shaping textiles* which consists in extruding molten plastic into pre-stretched fabric using additive manufacturing technologies. This printing-on-fabric technique recently gained popularity as a low-cost fabrication method for designing *deployable structures* which can pop out of shape when the fabric is released.

The morphing behavior of this composite material is due to a combination of intrinsic and extrinsic geometric transformations, the deposited plastic is both frustrating the metric of the textile and forming a bilayer with its substrate. We leverage these observations to design metamaterials which exploit both of these effects: the metric of the fabric is modulated by the density of printed patterns, and the amount of bending is controlled by the thickness of the printed plastic through the bilayer effect.

We show how these metamaterials can help for the design of self-actuated, lightweight structures by providing two separate kinds of design tools: forward design, or form-finding tools which aim at predicting the final deployed shape from a given printed layout, and an inverse design tool which starts from an input shape and find the optimal metamaterial parameters (thickness and density) to reproduce best the target shape.

Keywords: self-shaping – 3D printing – form finding – metamaterial – bilayers

CONTENTS

Contents	xi
1 Introduction	1
1.1 Context	1
1.2 Contributions	4
1.3 Outline	7
1.4 Publications	7
2 Background and related work	9
2.1 Computational design of deformable materials	10
2.2 Self-shaping materials	16
3 Creating self-shaping architectural models	27
3.1 Introduction	27
3.2 Physical Model	29
3.3 Fabrication	39
3.4 Evaluation	40
4 Bilayer shell simulation of self-shaping textiles	45
4.1 Introduction	45
4.2 Background	49
4.3 Material model	51
4.4 Measurements and fitting	54
4.5 Preliminary results and validation	59
4.6 Discussion and limitations	61
5 Inverse design of self-shaping textiles	63
5.1 Introduction	63
5.2 Related work	68
5.3 Programming intrinsic and extrinsic curvature	69
5.4 Ribbon layout	80
5.5 Fabrication	83
5.6 Results and discussion	86
6 Conclusion	91
Bibliography	95

CHAPTER 1

INTRODUCTION

1.1 Context

Advances in computer numerically controlled (CNC) machinery, jointly with the development of computational tools that permit the design and processing of freeform geometry allowed to reach an unprecedented level of complexity and customizability for the design and fabrication of objects. However this complexity of form comes with complexity in the manufacturing process: subtractive technologies such as CNC milling generate a lot of waste and are limited in the shapes they can make, while additive manufacturing needs a lot of support material to deal with overhangs. Now that freeform geometry is obtainable, we should strive for *efficient* ways to manufacture complex geometry. In that context, *deployable structures* offer an interesting alternative, as they can be manufactured in simple configurations and then bent or otherwise deformed into shape. Besides the easier fabrication process, flat-deployable structures have a compact form-factor which gives them benefits in the most mundane storage and transport situations as well as the most exotic ones such as space exploration. To manufacture an object flat and then deform it to match a specific shape requires a precise understanding of both the physics of the material in question and the geometry of the deformation, which will be detailed in the following chapters.

Among deployables, self-shaping structures represent an interesting case. They do not require any kind of actuation force to be deployed but instead morph as a reaction to an energy input which can be heat, light, moisture, a magnetic field or the energy stored in a pre-stretched substrate. Since their morphing is driven by a physical process (as opposed to manual deformation), they can be actuated more easily and in a more consistent manner. This mechanism can also prove very useful in situations where it is not possible to actuate the shape by pushing or pulling points on the surface, either because the context makes it difficult or because moving individual points on the surface will not

achieve the desired result.

There are many different ways of programming curvature into 2D sheets so that they self-actuate into a 3D shape. I investigated a technique to create *self-shaping textiles* which consists in depositing patterns of molten plastic into a pre-stretched fabric layer. The plastic bonds with the fabric layer and cures into a rigid material which will prevent homogeneous contraction of the fabric when it is released and favor buckling into 3D shapes instead (see Fig.1.1). Compared to other materials that can be used to program self-morphing sheets such as liquid crystal elastomers (LCE) [3] or swelling gels [63], self-shaping textiles have the advantage of being comparatively inexpensive and easy to fabricate. The only equipment needed is a stretchy type of fabric (e.g. spandex) and hobbyist 3D printing hardware to deposit the plastic, which makes this technique accessible to non-experts and opens the door to a wide range of creative practices and shape experimentations. In particular, this allowed architecture and design teachers to organize workshops in fab labs and similar types of facilities where students were able to

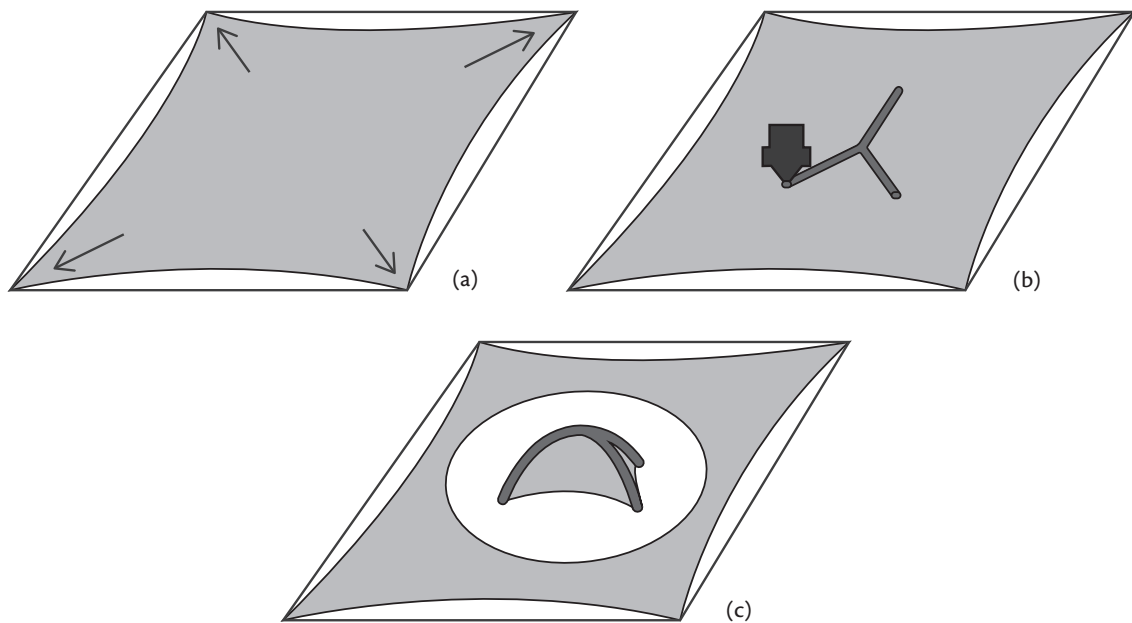


Fig. 1.1: Overview of the printing-on-fabric technique: (a) a square piece of fabric is stretched on the buildplate of a conventional (FDM) 3D printer, (b) the printer prints a shape as it would normally do, (c) the shape buckles out of the plane after being cut out from the pre-stretched fabric.

quickly prototype self-actuated shapes [2, 37, 69].

1.2 Contributions

The work presented in this thesis takes both a *geometric* and *design* point-of-view to deal with the computational design and fabrication of self-shaping textiles using the aforementioned printing-on-fabric technique.

Geometry. From a differential geometric standpoint, the deployment of self-shaping textiles can be analyzed through either an *intrinsic* or *extrinsic* perspective. In the intrinsic viewpoint, we are interested in how local quantities such as geodesic lengths and angles evolve on the surface. These lengths and angles can be measured by an object called the *metric* or the *first fundamental form* of the surface $I_p(u, v) = \langle df_p(u), df_p(v) \rangle$, which is the scalar product of the pushforward of two vectors u and v onto the tangent plane of the surface at a point p . The first fundamental form is a very powerful object, even though it gives no information about how a surface is embedded in 3D space, it uniquely identifies the *Gaussian curvature* K at every point.

However, in the general case, knowing the first fundamental form (or the Gaussian curvature) of a surface is not sufficient to uniquely determine it because there might be several possible surfaces that are equivalent up to an isometry (see [Figure 1.2](#)). This is why the *extrinsic* perspective is important as well: it can be compelling to only think of deployable materials in terms of how they stretch or expand and how this translates to curvature, but ultimately intrinsic notions give no information as to how a given surface is supposed to be embedded, or in our case, how it is supposed to *deploy* in 3D space. *Extrinsic* quantities, – such as the normal curvature κ_N of a curve, the mean curvature of a surface H , or its *second fundamental form* $II_p(u, v) = -\langle dN_p(u), df_p(v) \rangle$ which gives information about how normals around p evolve through the differential dN_p – help disambiguate scenarios where a surface with a given metric can have multiple embeddings.

In this thesis, we leverage both *intrinsic* and *extrinsic* effects. By carefully controlling the density of printed plastic patterns, we can modulate the ability of the fabric to retract back to its original size, and therefore control the underlying metric of the deployed surface. This allows to program Gaussian, or *intrinsic*, curvature into the deployable textile. To bias the possible solutions to a specific embedding, we control the normal curvature of the printed curves by leveraging a *bilayer effect*: since the fabric below the printed curves

is pre-stretched and the plastic material prevent contraction, the plastic-fabric bilayer will bend with a curvature depending on the thickness of printed plastic. This control on the *extrinsic* curvature guarantees a theoretical uniqueness of the deployed surface, as there can exist multiple surfaces with the same Gaussian curvature but there is only one surface (up to a global rotation and translation) with the same Gaussian and mean curvature [33, p.239].

Computational design. This thesis is about developing computational tools for the design of self-shaping textiles. We envision two possible workflows: a *direct* one and an *inverse* one. In a direct workflow, the user designs a specific pattern and then prints it on stretched fabric. If the deployed shape does not have the expected appearance, the user then has to start all over again. In that workflow, it can be useful to be able to perform *form-finding*, i.e. to computationally predict what the deployed shape will look like before printing it. This allows for faster and less wasteful iterations loops, because the user does not have to wait for the final printed object to decide whether they want to discard it or not (Fig. 1.3, left).

The inverse workflow is a bit different. Instead of specifying a pattern to be printed and discovering what kind of shape emerges from that pattern, the inverse process consists in first specifying a target shape and then try to find out which kinds of patterns can be suitable to obtain such a shape. For example, we can imagine that designers Guberman and Clopath [51] proceeded in some kind of inverse workflow when they designed their self-shaping shoe: they probably knew beforehand that they wanted to create a shoe, and then they figured out which curve layout to print so that the deployed shape looks like a shoe in the end. In that case, a computational tool which finds an optimal curve layout to be printed for a given shape can be of great help because it seemingly nullifies

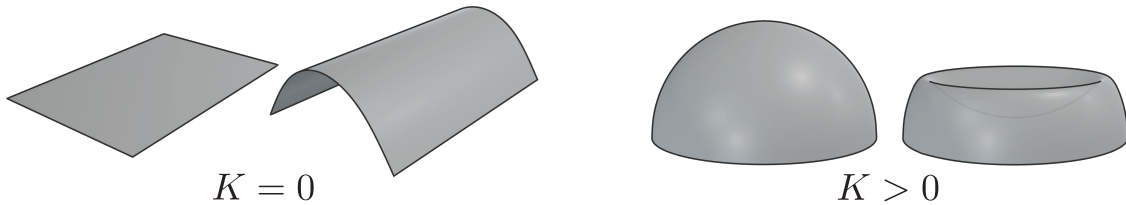


Fig. 1.2: Two shapes with the same Gaussian curvature can be bent differently, illustration with a developable surface ($K = 0$) and a sphere with positive gaussian curvature ($K > 0$) which can be bumped inwards in a similar way to a deflated football.

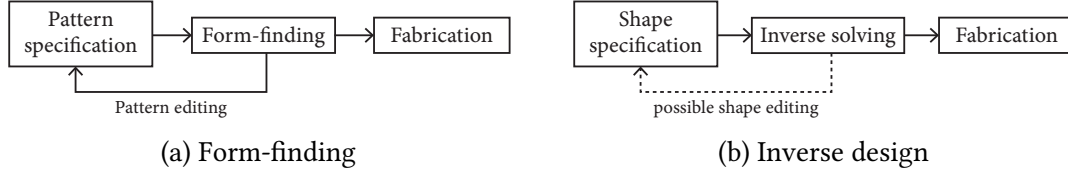


Fig. 1.3: Illustration of the two design workflows explored in this thesis.

the need for an iteration loop, even though in a real-life scenario the designer might not be completely satisfied with the result and may want to modify the input shape so as to generate different printing patterns (Fig. 1.3, right).

These two workflows correspond to different design cases because they operate from different starting points in the design process. In the direct workflow, a user starts from a pattern to be printed and then iterates on that pattern until a satisfying result is found, this may be useful in early stages of the design process when the user tries to explore the possible shapes they can obtain, or for artists who wish to enforce specific aesthetics to the printed curve pattern. In the inverse workflow the user starts from the shape to be reproduced, and obtains a pattern to be printed as a result of an optimization process, the user can then modify the input if the resulting pattern from the optimization is not satisfying.

In this thesis we address both design workflows, and towards this goal we tackle two main questions. First, is it possible to accurately predict the behavior of these printed-on-fabric composite materials? The answer is not obvious because textiles can be extremely diverse in terms of material response, and the process itself is imprecise with unknown variables such as plastic adhesion with the fabric, or over- or under-extrusions changing the result. To simplify this problem, we first study a particular pattern composed of tilings of stars which is simple enough so that reduced order simulation models work well to reproduce it, expressive enough so that it is possible to create varied shapes. We then try to predict the final shape of more general, arbitrary patterns printed on stretched fabric, by modeling the precise anisotropic behavior of the pre-stretched fabric on one hand, and the bilayer nature of the plastic-fabric composite material on the other.

The second question is can we inverse the design process and find a specific pattern that, once printed on stretched fabric, will morph and reproduce a given target shape? Inverse problems are in general ill-posed and this one is no exception: multiple printed patterns

could result to the same shape, while other geometries might not be reproducible at all by self-shaping textiles. Instead of optimizing for any kind of printed pattern, we focus on parallel ribbon patterns which allow to control both *intrinsic* curvature through their widths and spacing, and *extrinsic* curvature because of the bilayer effect.

1.3 Outline

The rest of the thesis will be as follows:

- Chapter 2 details the work which has inspired and is most related to this thesis.
- Chapter 3 presents a (direct) design tool for fabricating lightweight architectural models based on a tiling of star patterns, with the dimensions (and hence physical properties) of the individual pattern elements varying over space. Users of this system design free-form shapes by adjusting the star pattern; our system then automatically simulates the complex physical coupling between the fabric and stars to translate the design edits into shape variations.
- Chapter 4 describes preliminary work towards extending the form-finding tool to a general-purpose simulation method that can predict the buckling behavior of a variety of printed-on-fabric designs besides star tilings. The full anisotropic, large-strain response of the fabric is reproduced and integrated in a shell simulator which models the *bilayer effect* caused by the differential compression at the interface between the plastic and fabric layers.
- Chapter 5 presents an inverse design tool that is able to reproduce target shapes with printing-on-fabric using a dense pattern of closely-spaced ribbons. The core of this 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.

1.4 Publications

Chapters 3 and 5 are based on the two following peer-reviewed publications:

- David Jourdan, Mélina Skouras, Etienne Vouga, Adrien Bousseau, 2021. Printing-on-Fabric Meta-Material for Self-Shaping Architectural Models. In *Advances in Architectural Geometry 2020*, pages 264–285.
- David Jourdan, Mélina Skouras, Etienne Vouga, Adrien Bousseau, 2022. Computational Design of Self-Actuated Surfaces by Printing Plastic Ribbons on Stretched Fabric. *Computer Graphics Forum (Proc. EUROGRAPHICS, conditionally accepted)*.

Chapter 4 presents preliminary results of an ongoing work that has not been published yet.

CHAPTER 2

BACKGROUND AND RELATED WORK

The study of self-shaping textiles is an interdisciplinary subject which requires knowledge in theoretical areas such as differential geometry, continuum mechanics, and more practical ones like digital fabrication and additive manufacturing. As such, the literature which inspired this thesis covers a wide range of domains such as computer graphics, architectural geometry, human-computer interactions (HCI), and soft matter physics. We organize this chapter in two distinct sections:

- Section 2.1 deals with the computational design of deformable structures and materials, approached from the point of view of computer graphics and architecture, fields which focus on efficient tools and representations. In this body of work we distinguish studies on the *forward design* of deformable shapes and the related problem of form-finding (2.1.1) from *inverse design* tools. While form-finding deals with the task of finding the rest shape of a given structure – which we perform in chapters 3 and 4 – inverse design tools aim at programming curvature into the structure of a deformable material, in order to obtain a specific geometry – which we tackle in chapter 5. In particular we review different optimization devices which make this feat possible.
- Section 2.2 focuses on deformable materials which do not need any external actuation to be deformed into shape, but rather deform themselves. The self-actuation can be driven by different environmental stimuli such as heat, moisture, or in our case, the residual stress of a stretched substrate, we explain key mechanisms used in many of these projects such as the bilayer effect. These self-actuated structures have been studied in the soft matter physics community which is interested in understanding complex behaviors at the intersection between chemistry, physics and geometry. This understanding leads to accurate fabrication methods in controlled environments ; meanwhile computer graphics and HCI communities have devel-

oped more practical applications. In particular, we finish by giving an overview of the research in self-shaping textiles.

2.1 Computational design of deformable materials

This thesis is a contribution towards the goal of designing deformable structures that can be fabricated flat and deformed into a complex, 3D structure. The need to manufacture 3D surfaces out of flat sheets of material has motivated significant research in various areas such as computer graphics, architectural geometry, and materials science. This section does not attempt to provide an exhaustive analysis of the work from these different fields, but rather highlights some of the similarities between their approaches and how they relate to the work that we have done.

2.1.1 Forward design and form-finding

The computational design of deformable materials in computer graphics is historically linked to the problem of modeling and representing shape. Advances in the geometric understanding of deformations lead to better representations and algorithms for modeling shapes in the computer as well as methods to fabricate them with a given material. We see a prime example of that link with *developable surfaces*, which are surfaces with zero Gaussian curvature $K = \kappa_1 \kappa_2$ – where κ_1 and κ_2 are the maximum and minimum normal curvature. Gauss’s theorem egregium [40, p.27–28]

shows that such surfaces are locally isometric to the plane. As such, developable surfaces are an appropriate representation for materials that can bend but cannot shear or stretch, like paper. Approaches for modeling developable surfaces can thus represent them as smooth geometric primitives such as splines [136], or use algorithms based on discrete differential geometry such as the discrete orthogonal geodesic (DOG) nets of Rabinovich et al. [104].

Other types of surface representations allow the modeling of different material constraints, for example *Chebyshev nets* represent wire-like materials which have the ability

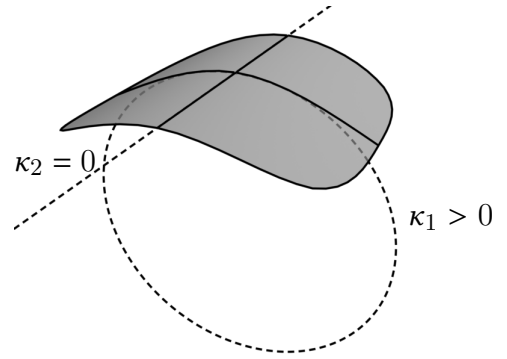


Fig. 2.1: A developable surface, κ_1 is the inverse of the radius of the osculating circle.

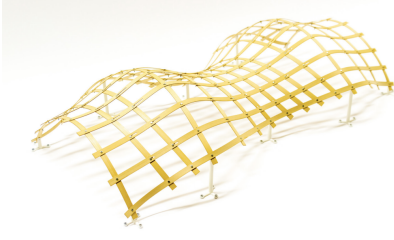


Fig. 2.2: Pillwein et al. [99]

to shear but not to stretch [39, 110] and can be used to model *elastic gridshells* [9] which are structures made of bent slender beams attached together to form a grid. A commonly-found feature of gridshells is that when they are made out of beams whose cross-section is anisotropic, the rod mechanics force them to be geodesics on the surface. This particular geometric property can be exploited to drive efficient algorithms for their design [99]. Once designed, deploying such structures can also be particularly challenging. A common approach is to assemble the grid flat and then lift it while fixing the beam endpoints to a prescribed boundary as was done for the ephemeral cathedral of Créteil [34] or the Solidays gridshell [11]. Otherwise, sparse actuation of the structure can be obtained by using a one degree-of-freedom mechanism with scissor-like hinges [94, 99, 129].

A family of surfaces that is particularly relevant to this thesis are *area-minimizing* or **minimal surfaces**, which are surfaces whose mean curvature $H = \kappa_1 + \kappa_2$ vanishes everywhere [33, section 3.5]. As their name implies, these surfaces have the property of minimizing their area under given boundary constraints. The numerical computation of such geometries has been a topic of interest for at least three decades [101] and has, until lately, remained challenging to solve for complex topologies and non-manifold configurations [58, 155]. But their interest is not only of a mathematical

nature, they are useful to model mechanical systems in which tension determines the final geometry, such as soap films whose geometry is effectively that of a minimal surface, or architectural surfaces such as *tensile structures* and cable nets [77] whose geometry is dictated by their internal stress. The architect Frei Otto famously experimented with soap films to prototype his tensile structures using a process called *form-finding*, which consists in experimenting with different boundary constraints to see a surface emerge out of the physical constraints [27]. In general, form-finding designates the process of finding the equilibrium shape of a given structure under a specific set of forces, it can be performed computationally to explore the shape space of surfaces in equilibrium under

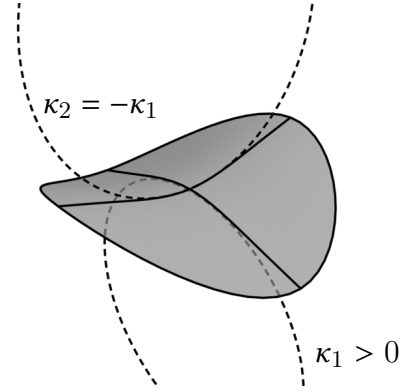


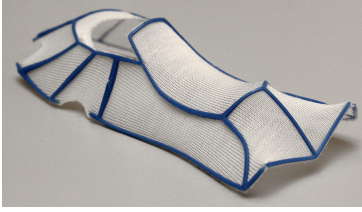
Fig. 2.3: A minimal surface.

internal stress (like tensile surfaces), self-weight (like catenaries and other funicular shapes) [81, 139] and bending-active shapes constrained by their endpoints [73].

Computational form-finding tools often try to solve a problem governed by the laws of physics. A popular approach called *Dynamic Relaxation* which is implemented in form-finding software such as Kangaroo Physics [86, 98] consists in integrating the equations of motion with an added damping factor to avoid parasite oscillations, the resulting motion is guaranteed to converge to a static equilibrium even if the motion itself is not physically realistic. In chapters 3 and 4, we employ a different strategy for form-finding self-shaping textiles. Instead of integrating the equations of motions we find the minimum of the potential energy of the system

$$\mathbf{x} = \operatorname{argmin} W(\mathbf{x}) \iff \nabla W(\mathbf{x}) = 0 \wedge \nabla^2 W(\mathbf{x}) \succ 0 \quad (2.1)$$

This problem is known as the variational approach in finite element analysis, we solve it using a modified version of Newton’s method where the energy $W(\mathbf{x})$ is used as a merit function [90]. For a more thorough overview of form-finding techniques we point the reader towards the work of Veenendaal and Block [149].

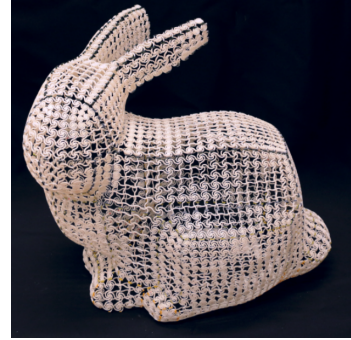


In their particular case of self-shaping textiles (see inset), Pérez et al. [95] tackle the form-finding problem by identifying that their designs live in a specific shape space, that of *Kirchhoff-Plateau surfaces* – area-minimizing surfaces which are enclosed by flexible rods. Notable designs com-

binning tensioned membranes and flexible rods, called Bending-Active Tensile Hybrid (BATH) structures [128], include the architectural work of Ahlquist and Menges [5] and Deleuran et al. [31]. In our case however, (chapters 3 and 5) the tensioned substrate is not completely enclosed by flexible rods. Therefore, the shapes we are able to create belong to a wider shape space.

Finally, it is relevant to mention that some structures have the ability to encode and control more general types of deformations. They are usually made out of repetitive, tileable elements that can contract or expand in a specific manner and thus induce intrinsic curvature by modulating the metric of the surface. Such structures are dubbed *metamaterials* as their deformation capabilities can be beyond those of the base material they are built with, and they can be defined by the properties of the individual element that is being tiled.

For example, Konaković et al. [66] consider triangular auxetic linkages where the opening angle between adjacent triangles dictates the resulting surface curvature, Chen et al. [24] augment this mechanism by using bistable tiles which can either be completely open or completely close, thus guaranteeing stability of the deployed shape. Malomo et al. [83] and Laccone et al. [74] consider repetitive spiral-shaped elastic rods where the amount of twist of the spiral controls surface bending (see inset figure). The tileable patterns we develop in chapters 3 and 5 can be considered to form a metamaterial as well.



In some cases, it is possible to control how the material will deform by carefully picking the right tileable elements that compose the metamaterial, and thus to create *programmable materials* which have their deformation behavior encoded within their structure, the process of designing such materials is called *inverse design*.

2.1.2 Inverse design

In the following, we distinguish between two kinds of inverse design methods. *Geometry-based* methods, solve the problem by studying its geometry, and use it to drive efficient algorithms. However, some problems may not have a nice mathematical structure which allows for simplifications. *Physics-based* methods approach the problem by using optimization machinery which poses static equilibrium as a hard constraint. Deformable materials which are made out of rigid parts, or that are constrained in their deformation are often well-suited for geometric methods. Flexible materials can also be programmed using a geometric approach (see e.g. [54]), and some rigid mechanisms can be suitable for physics-based optimization approaches (see e.g. [41]). It is also possible to combine a geometric approach with physics-based optimization [93, 106].

Physics-based

The physics-based optimization approach to inverse design consists in solving the following problem:

$$\begin{aligned} & \min_{\mathbf{x}, \mathbf{p}} E(\mathbf{x}) \\ & \text{subject to } f(\mathbf{x}, \mathbf{p}) = 0 \end{aligned}$$

where \mathbf{x} corresponds to geometry variables, \mathbf{p} are design variables (e.g. the Young's modulus in elastic parameter optimization [158]), $E(\mathbf{x})$ is the energy to optimize for (usually the distance to a target geometry possibly with added regularizers) and $f(\mathbf{x}, \mathbf{p})$ is the sum of the forces, which typically depends on both the geometry and the design parameters.

The constraint is usually nonlinear, so this problem is non trivial to solve and requires iterative methods with good initializations to avoid difficult local minima. As an example, Skouras et al. [125, 126] solve this problem with the Augmented Lagrangian Method (ALM) which iteratively updates lagrangian multipliers without having to introduce them in the linear systems, thus avoiding a significant increase in the number of variables compared to the traditional method of Lagrange multipliers (see [90, Chapter 17] for more details). Skouras et al. [127] use Sequential Quadratic Programming (SQP) to find the optimal layout of flat panels such that, when assembled, they form a balloon resembling the input shape.

An alternative to solving this problem using nonlinear constrained optimization tools is *sensitivity analysis*. Sensitivity analysis reformulates the constraint $f(\mathbf{x}, \mathbf{p}) = 0$ into an implicit function $\hat{\mathbf{x}}(\mathbf{p})$ which locally approximates the manifold defined by the constraint. Given variables $\mathbf{x}_0, \mathbf{p}_0$ such that $f(\mathbf{x}_0, \mathbf{p}_0) = 0$, we can write

$$f(\hat{\mathbf{x}}(\mathbf{p}), \mathbf{p}) \xrightarrow{\mathbf{p} \rightarrow \mathbf{p}_0} f(\mathbf{x}_0, \mathbf{p}_0) + \left(\frac{\partial f}{\partial \mathbf{x}} \frac{\partial \hat{\mathbf{x}}}{\partial \mathbf{p}} + \frac{\partial f}{\partial \mathbf{p}} \right) \Delta \mathbf{p} = 0 \iff \frac{\partial \hat{\mathbf{x}}}{\partial \mathbf{p}} = - \left(\frac{\partial f}{\partial \mathbf{x}} \right)^{-1} \frac{\partial f}{\partial \mathbf{p}}$$

The derivative $S = \frac{\partial \hat{\mathbf{x}}}{\partial \mathbf{p}}$ is called the *sensitivity matrix*, and can be used to optimize the geometry $\hat{\mathbf{x}}$ as a function of the design variables \mathbf{p} . Sensitivity analysis is often used in conjunction with the *adjoint method* which consists in avoiding a costly matrix inversion by exploiting the associativity of matrix multiplication when computing the gradient of

the energy: $\frac{\partial E}{\partial p} = \frac{\partial E}{\partial x} \left(- \left(\frac{\partial f}{\partial x} \right)^{-1} \frac{\partial f}{\partial p} \right) = - \left(\frac{\partial E}{\partial x} \left(\frac{\partial f}{\partial x} \right)^{-1} \right) \frac{\partial f}{\partial p}$ which amounts to simply solving a linear system.

Sensitivity analysis has been used for the design of objects made of flexible rods [83, 95, 96], silicone inclusions [159], polystyrene sculptures made with hot-wire cutting [35], objects weaved with curved ribbons [106] and inflatable structures [93]. Like other physics-based optimization techniques, its main drawback is that it requires simulating the entire object at every iteration which may be computationally expensive and prevent interactivity in the design process. In many cases, exploiting the *geometry* of the problem allows to derive more efficient algorithms.

Geometry-based

Because *developable materials* are only allowed to bend, they are often well-suited for geometric approaches. Tachi [134]’s Origamizer is a method to compute a pattern of folds on a sheet of paper so that when it is folded, the paper will resemble the input polyhedral surface. The idea of the method is that the excess of material can be tucked inside the folds when it is needed to create curvature, a process that has been used empirically before by Ron Resch [107]. The algorithm was proven to work on any polyhedral surface [32]. In contrast to origami where the practitioner is only allowed to fold the material, kirigami [60] allows for cuts, which enables less wasteful approaches for creating architectural geometries.

A different family of approaches start from a smooth surface (or a discretization of it) and optimize for some metric of developability such as the presence of hinges on vertex stars [133], the shape operator (i.e. the Hessian in isotropic geometry) being rank-deficient [116], or the Gauss image (the map of normals to the unit sphere) being one-dimensional [17]. Some methods are explic-

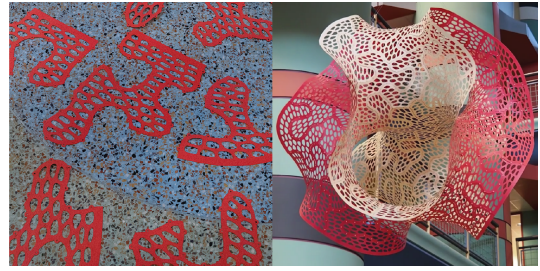


Fig. 2.4: Fabricating a doubly curved surface (right) by assembling planar patches (left) [80].

itly made to guarantee fabricability, such as the work of Ion et al. [57] based on the discrete orthogonal geodesic (DOG) framework. The method of Sharp and Crane [119] optimizes the shape of cuts in arbitrary surfaces such that they can be made from planar patches

with the least distortion possible, as showcased by Louis-Rosenberg and Rosenkrantz [80].

Shapes made out of assemblies of curves are also well-suited for geometric approaches, the case of *bending-active structures* that I mentioned in the previous section is a good example. Pillwein and Musialski [100] identify that curves in a gridshell correspond to geodesics, and devise an efficient way to compute them while optimizing for the orientation of the structure. Hafner and Bickel [54] devise a geometric criterion for all elastic ribbons in equilibrium and use it to optimize the cross-section of such ribbons to match a given curve input. A specific case relates to gridshells where the elements are *weaved* together. In this case they are not only geodesics but form a n -rotationally symmetric (n -RoSy) directional field. Vekhter et al. [150] identify geodesic foliations to be the proper mathematical framework for dealing with weaved straight ribbons, and develop an efficient algorithm to compute such foliations.

Closest to our inverse design applications (Chapter 5) are methods based on a *parameterization* approach. Konaković-Luković et al. [68] leverage the fact that the morphing of their metamaterial from the flat plane to a 3D surface preserve angles to use a conformal mapping tool to optimize for the size of elements in their triangular auxetic linkage so



Fig. 2.5: left: flattened auxetic pattern, right: projected pattern onto the target surface [68].

that the structure automatically takes the prescribed shape upon actuation by gravity or pneumatic deployment. Chen et al. [24] also use a conformal map to compute an isotropic scale factor for each individual cell of their metamaterial and match it to a precomputed cell shape which has the desired bistable behavior. The Geodesy+ tool [50] computes an anisotropic parameterization, because the contraction mechanism they exploit favors one direction (along the extrusion path) over the other, this parameterization is then exploited to create self-rising shapes.

2.2 Self-shaping materials

Self-shaping materials have the property of not needing any external actuation force to be deformed, instead their shape changes as a result of a variety of stimuli such as

temperature, moisture content, internal air pressure or tensile stress. In this section I give an overview of some of the mechanisms that make self-shaping materials possible.

2.2.1 Geometric description

In this section I make the distinction between two kinds of mechanisms to create self-shaping surfaces. The first ones act on the lengths of the surface to create Gaussian curvature and are called *intrinsic* mechanisms because they do not depend on how the surface is embedded in space. The second kind act on *extrinsic* notions of curvature such as bending by way of bilayer effects.

A general mechanism for self-actuation that was identified by Sharon and Efrati [118] is that of *metric change* meaning that some materials can locally expand or contract upon a specific trigger mechanism. If this contraction or expansion is not uniform across the whole geometry, it will trigger a change in curvature. Therefore, by changing the *metric* (also known as the first fundamental form) which is the bilinear operator defining distances on a surface, we can change the curvature of the surface. I previously mentioned Gauss' *theorema egregium* which states that the Gaussian curvature of a surface does not depend on how the surface is *bent* in space, which is why we consider that Gaussian curvature is an *intrinsic* property of surfaces. However, the link between the metric and Gaussian curvature may be more apparent if we introduce a corollary known as the Bertrand-Diguet-Puiseux theorem [15, 131, p.145] which states that Gaussian curvature can be expressed as the limit difference between circumferences, (or similarly, areas) of a circle in the plane and a (geodesic) circle of the same radius on the surface:

$$K = \lim_{r \rightarrow 0} 3 \frac{2\pi r - C(r)}{\pi r^3} = 12 \frac{\pi r^2 - A(r)}{\pi r^4}, \quad (2.2)$$

this clearly shows how Gaussian curvature is dependent on how the lengths on a surface differ from the lengths on the plane. If the circumference of the geodesic circle is bigger than the circumference of a flat circle, the Gaussian curvature will be negative, on the contrary if the circumference of the geodesic circle is smaller, then Gaussian curvature will be positive. Even though this formula is only true for infinitesimal disks on a surface, we can still see how the same principles can be applied on finite areas to create self-shaping surfaces.

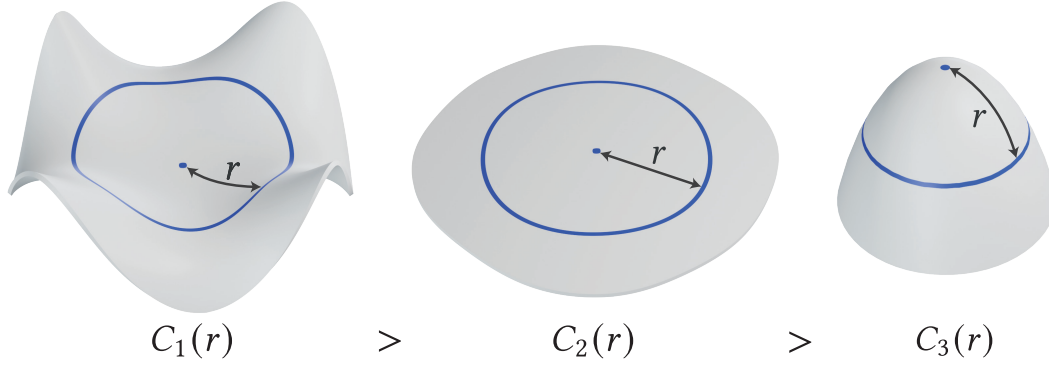


Fig. 2.6: Illustration of the Bertrand-Diguet-Puiseux theorem: the three circles have the same (geodesic) radius, but different circumferences due to the Gaussian curvature of the surfaces they are drawn on.

For example, Kim et al. [63] combine two materials with different swelling rates using a halftoning pattern, a different density of dots meaning a different swelling rate (see inset). Since the circumference is mostly covered by red dots which have a low swelling factor while the interior contains a black material with a high swelling factor, the area of the disk will increase faster than its circumference, thus creating positive Gaussian curvature.

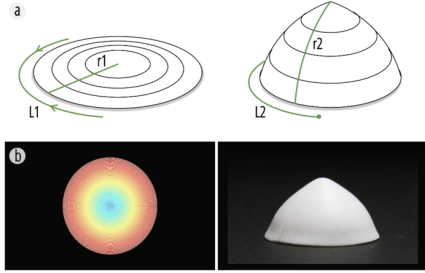
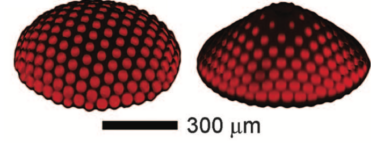


Fig. 2.7: Geodesy [49]

Some morphing materials [42, 49] leverage a different mechanism where the material has been deposited along specific pathways and shrinks at different rates along and across the printing path, which is particularly useful to create this difference between geodesic radius and circumference. In particular the Geodesy project [49, 50] focuses on shells printed along spiral pathways whose circumference shrinks faster than their radius, creating positive Gaussian curvature (see inset). A similar mechanism is at play on individual printed stars in Chapter 3, the star branches constrain the circle radius to stay constant while the circumference shrinks.

The difference of shrinking between different directions is an example of an *anisotropic* morphing behavior that can be used effectively to create self-shaping surfaces [45]. In particular, this mechanism is at play with nematic elastomers [3] where the orientation

of liquid crystals influences the swelling behavior and can be programmed by using a parameterization approach similar to ARAP [79]. We take inspiration from nematic elastomers and the general problem of anisotropic morphing to build our own ARAP-like parameterization algorithm for self-shaping textiles in Chapter 5.

An *extrinsic* mechanism that we exploit in chapters 3 and 5 is called the bilayer effect. It is the result of binding two layers, with at least one being an *active* layer which is either expanding or shrinking, the mismatching lengths between the two layers cause the initially planar assembly to bend out (Figure 2.8). This mechanism has been studied at least since Timoshenko [141] who predicted the curvature of a bilayer ribbon made out of two metals with different thermal expansion coefficients, we show how their formulation is equivalent to the one we computed for plastic-fabric composites in Chapter 3.

Several papers exploit this mechanism to create self-actuated rods [18, 109, 152], either by binding two layers of different materials with different properties, or by binding layers of the same material a different orientations, this works if the material in question has an anisotropic morphing behavior. Bilayer, self-shaping rods can be combined together to create more complex structures, such as self-shaping meshes [153] (see inset), or in our case, self-shaping rods attached to a pre-stretched substrate (chapters 3 and 5)

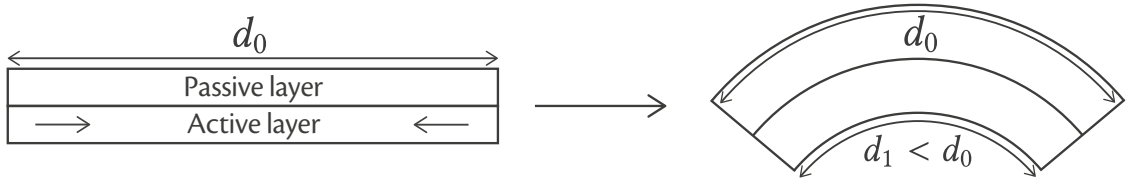
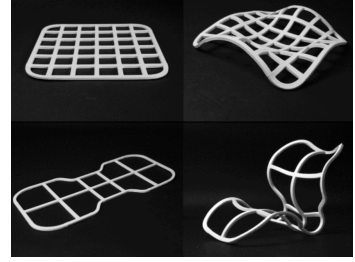


Fig. 2.8: Bilayer effect: the combination of a passive layer resisting deformation and an active layer which is either expanding or shrinking, causes the whole assembly to curve so that one layer becomes smaller than the other one. The curvature of the resulting equilibrium configuration can be computed as a function of the thicknesses of the layers, their Young's moduli, and the amount of expansion or shrinkage the active layer is going through.

More complex bending behavior can also be achieved with the bilayer effect so as to obtain a wide variety of developable surfaces [8, 144]. This mechanism can be used in conjunction with metric change: if both layers are *active*, then the metric of the whole assembly is going to change [146]. In fact, apart from origami techniques, metric change is *necessary* to create double curvature, as the bilayer effect itself can only *bend* the surface and thus cannot change the Gaussian curvature.

Equations for general bilayer shells have been formulated by Pezzulla et al. [97] and van Rees et al. [147]. Compared to standard finite element methods where a strain is computed between the current state and a stress-free rest state, it is more complex to simulate self-shaping bilayer shells because there may not exist a shape in which the shell is completely stress-free. Instead it is possible to compute stress-free first and second fundamental forms separately even if there might not exist a surface with such first and second fundamental forms, we use the formulas of van Rees et al. [147] in Chapter 3 to derive the curvature of a printed-on-fabric bilayer ribbon.

2.2.2 Materials

Self-shaping surfaces can be created using different materials and trigger mechanisms, in the following section I review some of them and classify them by their trigger mechanism. Some materials morph when receiving a certain quantity of energy, while some others were already storing energy inside and the trigger merely permits the release of the stored energy, which is the case of self-shaping textiles that I explore in this thesis.

A commonly heard term when talking about self-shaping materials is *4D printing* [140]. Unfortunately, this expression has no precise or commonly agreed on definition. In this thesis, the term 4D printing will be used only when referring to self-shaping objects that have been created using additive manufacturing technologies such as fused filament fabrication (FFF), this excludes techniques employing a pre-stretched substrate.

Many 4D printing methods rely on *heat* to trigger the morphing [8, 18, 49, 50, 144, 152, 153]. They exploit a warping effect that happens when the thermoplastic material that has been used to print the object is heated above its *glass transition temperature* (usually between 60–80°C). When a thermoplastic material is deposited using the common FFF technique, the polymer chains are initially aligned along the printing path. Heating the material allows the chains to rearrange along random directions, making the material

denser and more compact. This phenomenon makes the material shorten along the printing path, which is useful to leverage the bilayer effect to make self-shaping rods [18, 152, 153] and shells [8, 144], it can also be used to trigger more general metric changes [49, 50]. An innovative method combines 4D printing and conductive materials to create tangible user interfaces [26], by controlling current through the material to cause it to heat and morph.

Other types of materials can morph in reaction to a change of temperature, such as the classic case of metals studied by Timoshenko [141]. Boley et al. [20] combined four different elastomeric inks with various expansion coefficients to create lattices which morph out of the plane when heated, and then cure at a high temperature and can stay in place when cooling down to room temperature. This property is particularly desirable as some materials do morph when heated but go back to their initial shape upon cooling down, which means the programmed curvature cannot exist at room temperature. This is the case of Aharoni et al. [3]’s work, who programmed curvature into Liquid Crystal Elastomers which expand anisotropically when heated but go back to their initial shape at room temperature. A way to circumvent this problem is to prepare the sample at a high temperature and then let it morph by cooling down, even if the mechanism is reversible the sample will hold its shape at room temperature. This is the case for *gel lithography* [63, 87] methods which program a pattern in a photo-crosslinkable material using light in a controlled environment at a high temperature and then let the material cool down in an aqueous solution at room temperature.

Tibbits [140] who coined the term 4D printing, explores a different mechanism. Using a multi-material 3D printer, they mix a passive material with a hydrophobic one which expands when in contact with **water** and use this technique to engineer self-folding materials. Wood is also known to expand when wet, since its swelling is anisotropic and follows the material fibers, it is possible to assemble wood bilayers whose fibers are at different orientations to create bending [48, 109]. It is even possible to use additive manufacturing technologies to effectively *3D print wood* and therefore have a higher level of control on the anisotropic swelling behavior [25]. Other materials can have interesting behaviors in reaction to humidity, for example Gladman et al. [42] consider hydrogels which have an anisotropic swelling behavior when immersed in water, the results of this experiment were successfully simulated by van Rees et al. [146].

More unusual trigger mechanisms have been explored, such as cooking, to create self-shaping pasta [138, 156] or other flour-based foods [137]. Auto-inflatables [157] which are structures that are inflated by a chemical reaction inside the membranes can also lead to interesting developments, as there is a whole body of work on programming curvature into inflatable structures [91, 93, 124].

2.2.3 Pre-stretched membranes

The last trigger mechanism for self-shaping surfaces is the most related to this thesis: using the tension of pre-stretched substrates. The idea is to locally bind some rigid material with a pre-stretched membrane which forms a bilayer where the rigid material has the role of a *passive layer* resisting deformation, upon release the membrane becomes an *active layer* which will contract as much as possible.

One way to create self-shaping mechanisms is by altering the pre-stretched substrate itself, either by depositing resin onto the membrane to locally stiffen it, or by integrating rigid elements such as rods within the fabric. Oxman and Louis-Rosenberg [92] deposit resin into a pre-stretched latex membrane, forming different patterns and experimenting with simple form-finding algorithms using the Processing environment. Ahlquist et al. [4] and Sharmin and Ahlquist [117] use pre-stretched machine knitted textiles locally reinforced with epoxy resin and experiment with simple base forms and heuristic simulation models. Aldinger et al. [7] sew stiff fiber reinforced polymer rods into a pre-stretched tulle mesh fabric, they are able to control the curvature of the result depending on the pattern the rods form and whether they are on the top side or the bottom side of the fabric.

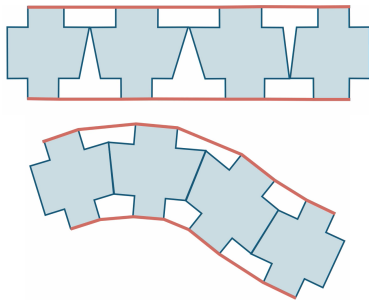


Fig. 2.9: CurveUps [53]

Another popular technique is to laminate a layer of stiff material onto the pre-stretched substrate. The layer itself has usually a sparse geometry to let the membrane contract and can be either 3D printed [1, 12, 21, 52, 53] or laser cut from materials such as wood [4, 12]. Some authors experimented with patterns forming a grid, for example Ahlquist et al. [4] glues laser cut Voronoi patterns onto a pre-stretched textile and examine their folding behavior, Agkathidis et al. [1] and Berdos et al. [12] print patterns following principal stress directions which are important for structural stability at a large scale. A problem

with this type of approach is that a dense grid will constrain the metric of the final geometry so that it will only be possible to obtain developable surfaces since the composite material can barely stretch or expand. Moreover, since the curves have an anisotropic cross-section, all the laminated lines are constrained to be geodesics on the surface. Another approach is to control the geometry of the material to obtain specific bending modes. Callens et al. [21] glue 3D printed structures with specific hinges on a latex membrane so that upon release the hinges make a 90° angle and the overall structure forms a triply periodic minimal surface. Guseinov et al. [53] propose a more general method to program curvature into pre-stretched membranes, which consists in ‘sandwiching’ 3D printed frustum-like elements between two rubber sheets, the specific geometry of the rigid elements allowing to control bending angles on the surface. [52] show how to also control the temporal morphing behavior of these structures by using heat as a second trigger mechanism.

Instead of laminating a given structure onto the pre-stretched substrate, it is possible to use a 3D printer to directly print onto the fabric material and the molten plastic will bond with the textile fibers upon cooling down. This technique has been also experimented for printing onto non-stretched textiles for composite applications, in particular there has been a lot of studies into the adhesion of 3D printed patterns onto textiles. See [123] for an overview on 3D printing on non-stretched fabric.

Directly 3D printing on stretched fabric to create self-shaping textiles has been used in many different applications such as to design shoes [51], architectural shell prototypes [6], circular shading panels [71], sound-absorbing panels (using tilings of stars similar to the one showcased in Chapter 3) [30] and even wearable tangible user interfaces [43], and for good reason: fabric is an ubiquitous and very versatile material, and has countless uses in fashion, sound absorption and architecture (see Fig. 2.10).

However, the morphing behavior of these assemblies is difficult to anticipate just by looking at the 2D printed pattern. The deployed shape is the result of a stress-minimizing process which can be simulated, therefore researchers experimented with form-finding tools to predict the effect of specific patterns. If some architects experimented with heuristic form-finding [28, 72], more systematic and physics-based methods have been developed as well: experiments on a simple rectangle shape have been done using the FEM package Abaqus [132], but the most notable and closely related to this thesis is the

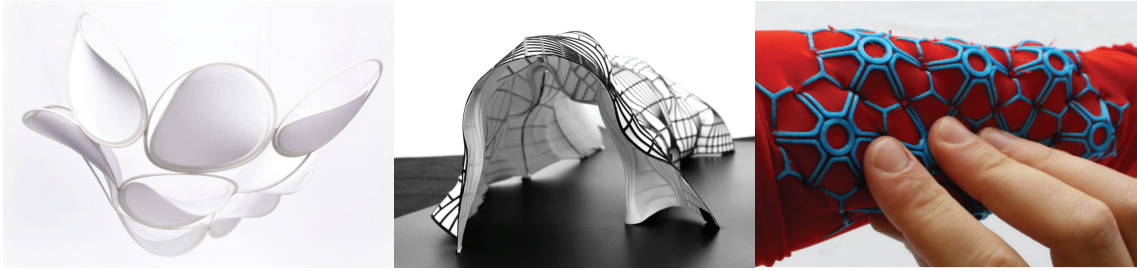


Fig. 2.10: Self-shaping textiles. From left to right: shading panels [71], architectural prototype [6], wearable user interface [43].

interactive form-finding program of Pérez et al. [95].

Perez et al.'s method focuses on Kirchhoff-Plateau surfaces which are piecewise-minimal-like shapes that appear when printing sparse networks of closed curves. If the network is more dense like a gridshell, the structure will become more constrained and only bending will be possible. This gives rise to developable or almost developable surfaces, as shown in the recent results by Agkathidis and Varinlioğlu [2]. But closed networks are not a requirement, in fact one of the strengths of this method is that by printing directly on the substrate we can easily print any pattern including disconnected ones. Indeed, experiments in workshops such as the one of Erioli and Naldoni [37] show great examples of that.

To control the metric in a more methodic way, it might be interesting to take inspiration from the whole body of work cited previously: even if they do not use the same materials, much of the same geometric principles can be applied. For example, a pattern of dots where the spacing between the dots varies can be a good way to control the metric in an isotropic, or *conformal* way [63], and indeed such a pattern was shown to work for self-shaping textiles [38].

Since the imprinted plastic layers form a bilayer with the fabric, it is convenient to exploit the bilayer effect to control the bending behavior of the surface. Starting with simple patterns such as parallel ribbons [72] or lamellæ [89], it is possible to identify which parameters have an impact on the resulting curvature. Parameters such as the amount of stretch of the fabric or the thickness of the plastic elements are known to control the bending behavior – for example, Kycia [70] exploited some of the capabilities of a 3D printer by varying the thickness of the printing pattern, this allowed them to locally modulate the amount of curvature along a printed element. More surprising findings are

that the number of parallel elements and their spacing also control the curvature: the more spaced the elements, the bigger the area of fabric which acts on the elements; and the bigger the number of elements, the larger the total force applied on the fabric [72, 89]. In Chapter 5, we exploit both the printed thickness and the ribbon spacing to modulate the curvature of our elements. Another interesting finding by Kycia [72] is how slender elements can either bend and roll into a cylinder, or make wrinkles depending on the thickness and the spacing of the elements. These two modes (wrinkling and bending) are related to the stiffness limits of bilayers and the theoretical limit between the two modes can be computed as a function of geometric and material parameters [145].

A limitation of using a 3D printer to deposit plastic on stretched fabric is that access to both sides of the substrate is difficult, which becomes limiting when one wants to exploit the bilayer effect: if only one side is covered by plastic pattern they will all be biased towards bending upwards. Christie [28] used a 6 degrees of freedom (DoF) robot arm to print both onto and under a pre-stretched textile, the printed shapes are not forming bilayers but are tridimensional mechanisms allowing some metric change as well as some bending.

CHAPTER 3

CREATING SELF-SHAPING ARCHITECTURAL MODELS

3.1 Introduction

Physical models are an important tool architects use to explore ideas and communicate them to clients and collaborators [36]. Architects have a long tradition of exploiting affordable and easy-to-assemble materials for fabricating small-scale prototypes, including cardboard, foam and wood cutouts. With the rise of maker spaces, technologies for milling, laser cutting, and 3D printing are increasingly accessible and powerful tools for architectural prototyping. Doubly-curved freeform surfaces, however, remain a significant challenge: even with modern additive manufacturing techniques, prototypes of thin curved structures cannot be easily fabricated without formwork, support structures, or other artifacts.

This chapter explores the use of 3D printing on stretched fabric to create self-shaping architectural models. This fabrication technique can be a powerful medium for architects who wish to explore new shape ideas, as was demonstrated by recent experimentations [2, 37, 70]. Our work follows in the line of several recent explorations of combining fabric under tension with networks of elastic rods to control shape [1, 95]. The resulting fabric and curve structure behaves as a so-called *Kirchhoff-Plateau Surface*, where the fabric forms minimal surface patches bounded by the rods. Yet, a wider variety of shapes are possible if one does not restrict to printing closed curves and instead uses dense repetitive patterns of plastic rods [37, 70]. Our main contribution is the use of dense 3-pointed star patterns for fabricating freeform structures with large-scale curvature variation. Our choice of 3-pointed stars is based on two important physical observations: first, the arms of the stars bend to form a small bump localized around the star, which contracts the surface locally. Varying the thickness of the stars offers control on the amount of contraction in their neighborhood. Second, since the pattern is formed of disconnected elements, the fabric contracts in-between these elements, bringing them

closer together, with the amount of contraction controlled by the local spacing of the stars. In both cases, if the amount of contraction varies spatially, it induces in the initially-flat fabric a new rest state with *non-Euclidean metric* [63, 67, 118], and the fabric bends in space to resolve the metric frustration. Due to the combination of these effects, our plastic pattern act as a *metamaterial*, whose local geometry dictates the overall shape. Varying the two degrees of freedom offered by the stars thickness and spacing allows us to achieve a variety of shapes, including cylindrical and doubly-curved surfaces with negative Gaussian curvature (Fig. 3.9). An additional benefit of our star-based approach is that the pattern enriches the surface with a texture, allowing architects to explore a design space that combines shape and appearance.

The precise amount of contraction within and between the stars in a star pattern is determined by the nonlinear interaction of the stiff plastic with the fabric substrate. To avoid the need for laborious trial-and-error with physical 3D-printed models, we present a form-finding tool that simulates the deployment of our printed-on-fabric metamaterials, allowing users to quickly iterate on virtual prototypes before investing time in fabrication. To perform our simulation, we introduce a physical model of plastic patterns printed on stretched fabric. Our model goes beyond existing ones [95] by accounting for the bilayer structure of the plastic-on-fabric assemblies and for the specific mechanical behavior of the structure at the boundary between the patterns and the fabric, especially near the star arms tips, which we found to be critical to accurately reproduce the shapes we target. We also detail our fabrication protocol used to achieve accurate, reproducible physical prototypes using a commodity 3D printer.

In summary, we make the following contributions:

- A description of the main physical phenomena that contribute to the emergence of curved surfaces when printing dense plastic patterns over pre-stretched fabric.
- A physical simulator of dense plastic patterns printed on pre-stretched fabric, allowing accurate reproduction of these phenomena.
- A form-finding tool based on this simulator, which we used to design and fabricate a variety of architectural forms based on a simple 3-pointed star pattern.

This chapter is mainly based on the following publication:

- David Jourdan, Mélina Skouras, Etienne Vouga, Adrien Bousseau, 2021. *Printing-on-Fabric Meta-Material for Self-Shaping Architectural Models*. In *Advances in Architectural Geometry 2020*, pages 264–285.

3.2 Physical Model

To fabricate our self-shaping textile models, we take a rectangle of thin fabric, pre-stretch it along each dimension by a factor s by pinning the fabric boundary to a wooden frame, place the fabric on the print bed of a conventional FFF 3D-printer, and print small three-pointed stars over the fabric (see Figure 3.6). The melted plastic deposited by the printer adheres to the surface of the fabric, resulting in strong bonding of the two materials to each other. After the plastic has cured, we remove the fabric from the frame, trim excess fabric, and pose the structure by pinning several points of the boundary to the ground. See Section 3.3 for full details of the fabrication process.

We parameterize the fabric by a rectangle Ω in the plane. Each star’s center is placed at the nodes of a regular hexagonal tiling of Ω , with distance d mm between neighboring centers, with the star’s three arms aligned to the symmetry axes of the tiling. Three scalar

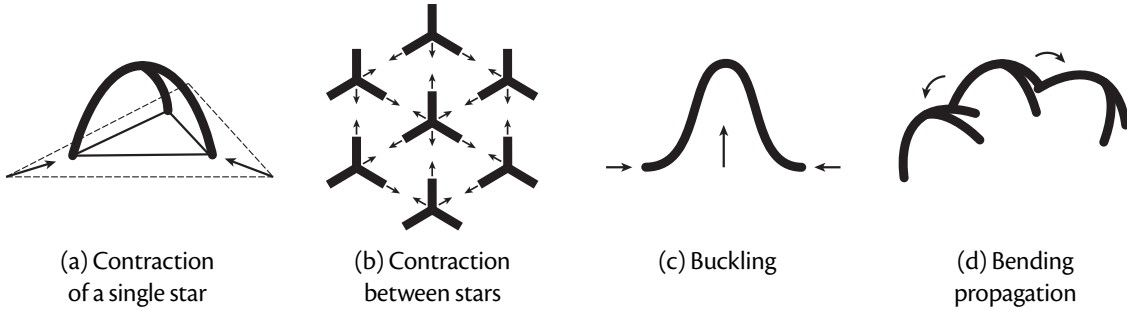


Fig. 3.1: Geometric intuition behind our approach, at several length scales. At the scale of a single star, the contraction of the pre-stretched fabric back to its rest dimensions is halted by the plastic star bonded to the fabric. The star arms buckle to form a small bump (a); neighborhoods of thicker stars contract less due to the star arms bending less. At the scale of a neighborhood of several stars, the fabric contracts unimpeded, since stars are not connected (b). A sparser star pattern with smaller stars (and hence more spacing between them) allows more contraction. At the scale of a large patch of metamaterial area, several factors control the surface shape: differential contraction due to variations in star thickness and spacing induces buckling of the surface to relieve metric frustration; boundary conditions can impose additional contraction and buckling (c); and if stars are laid out in a regular pattern, there is global coupling in how each star breaks symmetry while buckling, introducing large-scale curvature (d).

sizing fields specify the design of the star pattern: $\ell : \Omega \rightarrow [0, 1]$ specifies the length of the star's arms at different locations on the fabric, with $\ell = 0$ indicating no star at all and $\ell = 1$ a star with arm lengths d (so that the star touches its neighbors); and $h, w : \Omega \rightarrow \mathbb{R}$ specify the thickness (in the direction perpendicular to the fabric and printing plane) of the stars and (in-plane) width of the star arms, both in millimeters.

To summarize, a star pattern design consists of a choice of:

1. fabric tension s and star spacing d , both global to the entire pattern;
2. three functions ℓ, h, w over Ω ; which encode variations in the star sizing;
3. boundary conditions for how the border of the pattern should be pinned to the ground after printing.

3.2.1 The geometry of the metamaterial

After the star pattern has been printed and the fabric is allowed to relax to static equilibrium, the surface buckles into a 3D structure with residual internal stress. Figure 3.1 illustrates how the choice of design parameters provides several means of lifting the resulting surface into controllable shapes. In the neighborhood of each individual star, the star arms bend to form a bump under the action of the fabric's compressive forces (Figure 3.1a). The size of this bump depends on the fabric tension s and the length and thickness of the star arms, which control the star's resistance to bending and thus final curvature. In between stars, the fabric contracts unimpeded (Figure 3.1b), by an amount proportional to the length of the star arms.

At a coarse scale much larger than that of an individual star, we can treat the metamaterial as a homogenized smooth surface without the bumps around each star. In this homogenized view, the effect of each bump is to change the surface area of a neighborhood of the star at equilibrium, where the spatially-varying amount of surface contraction depends on the thickness, width, and length of the stars. Therefore ℓ, h, w equip the homogenized surface with a rest state described by a non-Euclidean metric [118]. In addition to changing the local surface area of the homogenized surface, the stars modify the surface's rest extrinsic curvature, since the stars are printed on top of the fabric (rather than embedded within it); in other words the metric of the metamaterial varies in the thickness as well as



Fig. 3.2: Our simulation tool (left) correctly predicts that a uniform star pattern, with free boundary, rolls up into a cylinder at static equilibrium (right). This complex behavior is due to the extrinsic curvature induced by the bilayer structure of the ribbon-and-fabric assembly, and the coupling of the bending of neighboring stars causing global symmetry-breaking in the pattern.

the curvilinear directions. The differential contraction described by this metric causes the surface to buckle out of plane, in order to exchange large amounts of stretching strain for slight bending strain (Figure 3.1c). This relationship between change of metric and buckling has also been exploited by related self-shaping fabrication technologies based on swelling [63] or auxetic linkages [67]. In our case, the precise relationship between the surface metric and the values of ℓ , h , and w depends on a complex physical coupling between the fabric and the printed plastic ribbons.

We also observed a coupling phenomenon between neighboring stars, where the bending of each individual star propagates to adjacent stars through deformation of the fabric in between (Figure 3.1d). It is unclear whether this behavior is a consequence of, or an additional effect independent of the induced non-Euclidean metric. This coupling leads to globally consistent symmetry-breaking in the surface, a phenomenon that is especially visible when we do not fix the boundary of the domain, since in this case the accumulation of local bending makes the entire surface fold on itself to form a tube, as shown in Figure 3.2.

Given the complexity of the physical phenomena involved, from local contraction of the surface to global propagation of bending, we propose a dedicated numerical simulation model to predict the shape that user-provided star patterns would take. In the following sections we present the different material models used, they have all been implemented in C++ and their implementation is available on GitHub [61].

3.2.2 Fabric's material model

We model the fabric as a plate discretized using constant strain triangles and edge-based bending hinges following the popular Discrete Shells model of Grinspun et al. [47]. The energy describing the behavior of the fabric is computed as a sum of a flexural term associated to bending, and a membrane term associated to stretch, it is optimized to find the static equilibrium configuration (see Section 3.2.4).

For the flexural term, we followed the implementation suggested by Tamstorf and Grinspun [135] and compute it as a sum over edges of the mesh:

$$W_B(x) = k_B \sum_i \frac{3\|\bar{e}_i\|^2}{\bar{A}_i} \left(2 \tan \left(\frac{\theta_i}{2} \right) \right)^2 \quad (3.1)$$

where θ_i is the dihedral angle, $\|\bar{e}_i\|$ is the initial length of edge e_i , \bar{A}_i is the sum of the initial areas of the adjacent triangles, $k_B = \frac{Eh^3}{24(1-\nu^2)}$ is the bending stiffness, h is the thickness of the membrane, E its Young's modulus and ν its Poisson's ratio. While bending forces are typically much smaller than membrane forces in Kirchhoff-Plateau surfaces and can be mostly neglected [95], we found that accounting for the finite flexural resistance of the material at the boundary between the stars and fabric was necessary to limit the tangent discontinuities at the boundary and to reproduce the global curvature of the surface that we observed on the real artifacts (see Figure 3.5).

Our structures exhibit high localized stresses and large deformations near the tips of the stars' arms, due to the arm tips bending and "digging into" the fabric substrate. Using a linear material model for the fabric causes elements to completely compress and cause so-called *altitude collapses* [135] which leads to division by zero in the Discrete Shells model. To prevent this problem, we use a neo-Hookean material model to approximate the nonlinear deformation of the fabric near the rod tips. The membrane energy contribution writes as

$$W_M(x) = \frac{h}{2} \sum_i \bar{A}_i \left(\mu(\text{tr}(F^T F) - 2 - 2 \ln J) + \lambda(\ln J)^2 \right) \quad (3.2)$$

where $\lambda = \frac{E\nu}{1-\nu^2}$ and $\mu = \frac{E}{2(1+\nu)}$ are the Lamé coefficients, F is the deformation gradient, and $J = \sqrt{\det F^T F}$. Note that this formula is slightly different than the equivalent one for volumetric elements.

Despite the orthotropic behavior exhibited by the fabric that we used, we noticed that in the case of this star tiling pattern it was sufficient to model it as an isotropic material.

3.2.3 Rods material model

We use the Discrete Elastic Rods (DER) model [13, 14, 75] for modeling the arms of the stars. This model is based on a reduced centerline representation, where a rod is represented as a strip of vertices with additional degrees of freedom to represent twist. For a rod with a rectangular cross-section with thickness w_n and width w_b , the energy contribution of a rod writes as:

$$W_R = \frac{1}{2} \sum_i \frac{EA}{8} \left(\frac{(\|e_{i-1}\|^2 - \bar{l}_{i-1}^2)^2}{\bar{l}_{i-1}^3} + \frac{(\|e_i\|^2 - \bar{l}_i^2)^2}{\bar{l}_i^3} \right) + \frac{2}{\bar{l}_{i-1} + \bar{l}_i} \left(EI_1 (\kappa_i^1 - \bar{\kappa}_i^1)^2 + EI_2 (\kappa_i^2 - \bar{\kappa}_i^2)^2 + \mu \frac{I_1 + I_2}{2} \tau_i^2 \right) \quad (3.3)$$

where $I_1 = \frac{w_n^3 w_b}{12}$, $I_2 = \frac{w_b^3 w_n}{12}$ are the geometric moments of inertia, $A = w_n w_b$ is the cross-sectional area, τ_i is the discrete twist associated with vertex i , κ_i^1 and κ_i^2 are the two components of the curvature binormal projected onto the local material basis. For more details, see [75].

Coupling. We enforce the coupling between the plate and the rods via collocation: all centerline degrees of freedom of the rod are also vertices of the plate. This strategy implies that the fabric mesh must have edges that align with the star arms: we first discretize the rods constituting each star (by choosing the resolution of the rod centerline) and then we generate the fabric mesh using constrained Delaunay triangulation [122]. Within a star, the rods should also be coupled together because they are rigidly linked at the center of the star.

Defining energies that allow for proper transfer of bending and twisting forces from one rod to the other is, in the general case, a challenging problem. Even though several models exist for computing connections between rods using within the DER framework [76, 96], they can be complex to implement and expensive to run. We instead exploit the 3-folded symmetry of stars and the fact that the rods mostly bend about their width axis: we compute twist and bending at the connection as if each star arm was split into two “copies” of the same rod with the same geometry and each was bonded to one rod copy on each of the two other arms (see Figure 3.3). The elastic energy contributed by the connection is then the sum of the three pairwise bending and twisting energies, weighted by $\frac{1}{2}$ since the cross-sectional area of each “rod copy” is half that of the full rod. This

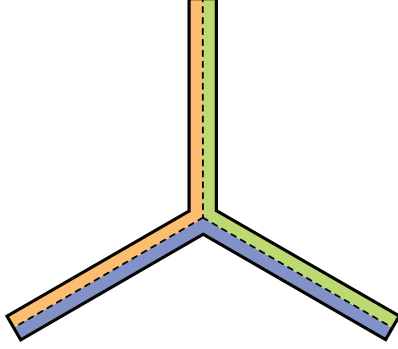


Fig. 3.3: To define a bending and twisting energy at the connection between three rods, we split each rod into two copies and divide the volume of the connection into three equal parts, here shown in different colors.

approximation gives the proper weighting only in the case where rods only bend about their width axis at the center connection, that is $\kappa_i^2 = 0$. In that case, both connection contributions are weighted by $I_1 = \frac{w_n^3 w_b}{24}$ and thus sum to the expected total.

Bilayer ribbon model. Simply colocating the vertices representing star arms to the thin shell model representing the fabric does not correctly capture the geometry of the ribbon-on-fabric assembly: in the printed assembly, the stars are *on top* of the fabric, while in the colocation model, rods are embedded *within* the fabric. Yet the bilayer structure of the assembly is precisely what causes the rods to bend in a privileged direction, because of the differences in stresses between the traction-free top surface of the ribbon and the compressive forces applied by the fabric to the bottom surface. We can easily observe this so-called *bilayer effect* on a flat ribbon printed on top of a strip of fabric of the same width: the structure will not stay flat when released and will consistently bend in the same direction (see Figure 3.4, left). Furthermore, the assembly also exhibits some plastic deformation: when we separate the ribbon from the fabric layer, the initially straight ribbon does not completely recover its original shape (see Figure 3.4, right) and has a non-zero rest curvature $\bar{\kappa}$.

Van Rees et al. [147] show that a bilayer shell made of two monolayers of respective Young's moduli E_1, E_2 , thicknesses h_1, h_2 , and where the strain-free rest state of each layer is given by a different metric (i.e. first fundamental form) a_{r1} and a_{r2} , is energetically equivalent to a monolayer with non-zero rest curvature. They give formulas for the rest first and second fundamental forms a_r and b_r of this equivalent monolayer (see [147, Supplemental material section S1.3.3]). To account for ribbon plasticity, we extend their formula to the case where each of the two layers in the bilayer have non-flat rest shape

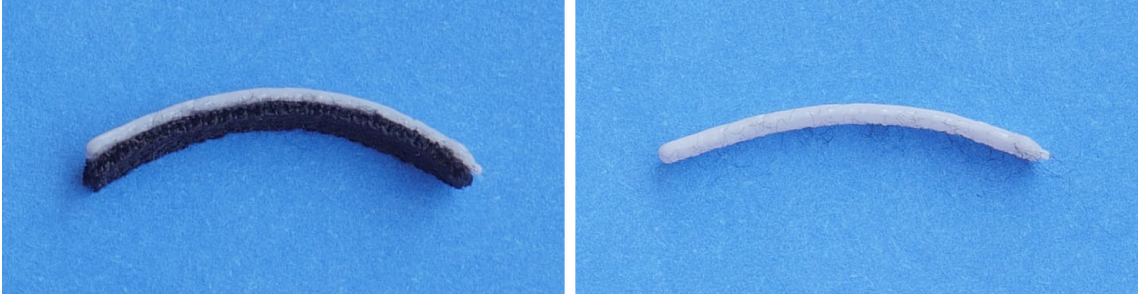


Fig. 3.4: A plastic ribbon printed on top of a strip of fabric. Due to the bilayer effect, the ribbon will naturally bend upwards (left). The plasticity of the material causes the upper layer not to recover its initially flat shape when detached from the fabric layer (right).

with the same second fundamental form \bar{b} .¹ Here we derive new formulas for a_r and b_r that account for this bilayer rest curvature.

To this end, we make the following ansatz:

$$\begin{aligned} a_{r1} &= a_r + \alpha_1(b_r - \bar{b}) \\ a_{r2} &= a_r - \alpha_2(b_r - \bar{b}), \end{aligned} \quad (3.4)$$

where α_1 and α_2 are factors to be determined so that the elastic energy of the effective monolayer and that of the bilayer agree up to a constant. Note that for any choice of these factors, Equation (3.4) is satisfied by setting

$$a_r = \frac{a_{r1} + a_{r2}}{2} - \frac{\alpha_1 - \alpha_2}{2(\alpha_1 + \alpha_2)}(a_{r1} - a_{r2}) \quad (3.5)$$

$$b_r = \bar{b} + \frac{a_{r1} - a_{r2}}{\alpha_1 + \alpha_2}. \quad (3.6)$$

Like van Rees et al. [147], we define the elastic inner product associated to a material with Poisson's ratio ν as

$$\langle A, B \rangle = \frac{\nu}{1 - \nu^2} \text{tr}(A) \text{tr}(B) + \frac{1}{1 + \nu} \text{tr}(AB) \quad (3.7)$$

and the elastic energy norm as

$$\|A\|^2 = \langle A, A \rangle = \frac{\nu}{1 - \nu^2} \text{tr}^2(A) + \frac{1}{1 + \nu} \text{tr}(A^2). \quad (3.8)$$

¹Note that in our case, only the plastic ribbon exhibits significant plasticity. However, assuming that the fabric has the same curved rest state as the ribbon helps simplifying the formulas. Since our fabric is highly flexible, we believe that the error that we introduce by doing so is negligible.

Letting a_c and b_c denote, respectively, the first and second fundamental forms of the midsurface of the shell in the current configuration, we can define the energy of the bilayer by

$$\begin{aligned}
 W_{BL} = & \frac{1}{2} \int_U E_1 \left[\frac{h_1}{8} \|a_{r_1}^{-1} a_c - I\|^2 + \frac{h_1^3}{24} \|a_{r_1}^{-1} (b_c - \bar{b})\|^2 \right. \\
 & \left. + \frac{h_1^2}{8} \langle a_{r_1}^{-1} a_c - I, a_{r_1}^{-1} (b_c - \bar{b}) \rangle \right] \sqrt{\det a_{r_1}} \, dx \, dy \\
 & + \frac{1}{2} \int_U E_2 \left[\frac{h_2}{8} \|a_{r_2}^{-1} a_c - I\|^2 + \frac{h_2^3}{24} \|a_{r_2}^{-1} (b_c - \bar{b})\|^2 \right. \\
 & \left. - \frac{h_2^2}{8} \langle a_{r_2}^{-1} a_c - I, a_{r_2}^{-1} (b_c - \bar{b}) \rangle \right] \sqrt{\det a_{r_2}} \, dx \, dy
 \end{aligned} \tag{3.9}$$

and that of the equivalent monolayer by

$$\begin{aligned}
 W_{ML} = & \frac{1}{2} \int_U \left[\frac{E_1 h_1 + E_2 h_2}{8} \|a_r^{-1} a_c - I\|^2 + \frac{E_1 h_1^3 + E_2 h_2^3}{24} \|a_r^{-1} (b_c - b_r)\|^2 \right. \\
 & \left. + \frac{E_1 h_1^2 - E_2 h_2^2}{8} \langle a_r^{-1} a_c - I, a_r^{-1} (b_c - b_r) \rangle \right] \sqrt{\det a_r} \, dx \, dy,
 \end{aligned} \tag{3.10}$$

where U is the parameterization domain of the shell midsurface parameterized by curvilinear coordinates (x, y) .

After plugging (3.4) in Equation (3.9), expanding the terms inside the norms, discarding all terms which do not depend on a_c nor b_c (and therefore do not change the equilibrium state), and equating all the remaining terms, we find that W_{BL} matches W_{ML} (up to a constant) when the following system of equations is satisfied:

$$\begin{aligned}
 \frac{E_2 h_2 \alpha_2 - E_1 h_1 \alpha_1}{4} &= \frac{1}{8} (E_2 h_2^2 - E_1 h_1^2) \\
 \frac{E_1 h_1^2 \alpha_1 + E_2 h_2^2 \alpha_2}{8} &= \frac{1}{12} (E_2 h_2^3 + E_1 h_1^3).
 \end{aligned} \tag{3.11}$$

Solving System (3.11) for α_1 and α_2 gives us

$$\begin{aligned}
 \alpha_1 &= \frac{4h_1^3 E_1 + 3h_1^2 h_2 E_1 + h_2^3 E_2}{6h_1^2 E_1 + 6h_1 h_2 E_1} \\
 \alpha_2 &= \frac{h_1^3 E_1 + 3h_1 h_2^2 E_2 + 4h_2^3 E_2}{6h_1 h_2 E_2 + 6h_2^2 E_2},
 \end{aligned} \tag{3.12}$$

which allows us to write

$$b_r - \bar{b} = \frac{a_{r1} - a_{r2}}{\alpha_1 + \alpha_2} = \frac{6h_1h_2E_1E_2(h_1 + h_2)}{h_1^4E_1^2 + 2h_1h_2E_1E_2(2h_1^2 + 3h_1h_2 + 2h_2^2) + h_2^4E_2^2}(a_{r1} - a_{r2}). \quad (3.13)$$

In our case, the quantities E_1 , h_1 and a_{r1} correspond to the *plastic* layer, and E_2 , h_2 and a_{r2} correspond to the *fabric* layer. The plastic material that we use is much stiffer than the fabric, i.e. $E_1 \gg E_2$, so we can further simplify this expression:

$$b_r - \bar{b} \approx \frac{6h_2(h_1 + h_2)E_2}{h_1^3E_1}(a_{r1} - a_{r2}). \quad (3.14)$$

We assume the plastic printed on top of the stretched fabric has no residual strain, so that the metric of the plastic is $a_{r1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. Since the fabric is stretched from rest by a factor s , $a_{r2} = \begin{pmatrix} \frac{1}{s^2} & 0 \\ 0 & \frac{1}{s^2} \end{pmatrix}$. We then have

$$b_r - \bar{b} \approx \frac{6h_2(h_1 + h_2)E_2}{h_1^3E_1} \begin{pmatrix} 1 - \frac{1}{s^2} & 0 \\ 0 & 1 - \frac{1}{s^2} \end{pmatrix}, \quad (3.15)$$

i.e.,

$$b_r \approx \frac{6h_2(h_1 + h_2)E_2}{h_1^3E_1} \begin{pmatrix} 1 - \frac{1}{s^2} & 0 \\ 0 & 1 - \frac{1}{s^2} \end{pmatrix} + \bar{b}. \quad (3.16)$$

The bending energy of the ribbon-on-fabric assembly will penalize deviation of the assembly's second fundamental form b_c away from b_r . Since the arms of our stars are much longer than they are wide, we neglect bending deformations in the width direction, and assume that the arms are at rest when the curvature κ in the longitudinal direction matches the normal curvature in that direction prescribed by b_r :

$$\kappa = \frac{6h_2(h_1 + h_2)E_2}{h_1^3E_1} \left(1 - \frac{1}{s^2}\right) + \bar{\kappa}, \quad (3.17)$$

where $\bar{\kappa}$ is the rest curvature of the plastically deformed rod in the longitudinal direction.

We can show that, in the case of $\bar{\kappa} = 0$, this result is consistent with previous models for bilayers such as the classic Timoshenko model for bilayer metal thermostats [141] which states that the curvature is given by

$$\kappa = \frac{6(h_1 + h_2)}{3(h_1 + h_2)^2 + h_1^2 + h_2^2 + h_2^3 \frac{E_2}{E_1} + \frac{h_1^3 E_1}{h_2 E_2}} \left(1 - \frac{1}{s^2}\right) \xrightarrow{E_1 \gg E_2} \frac{6h_2(h_1 + h_2)E_2}{h_1^3E_1} \left(1 - \frac{1}{s^2}\right)$$

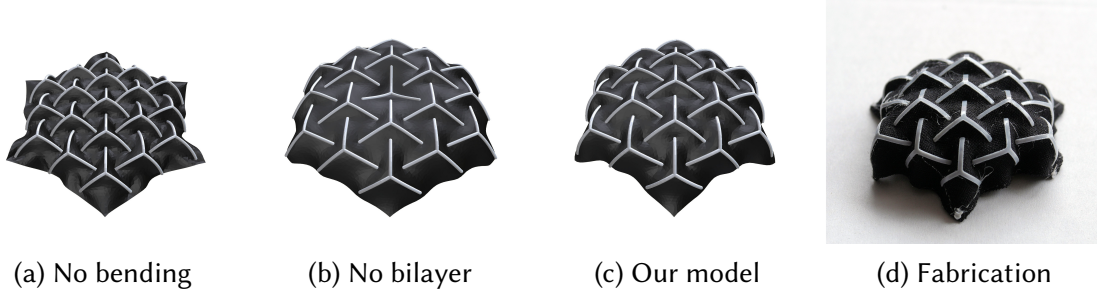


Fig. 3.5: Impact of critical components of our model when simulating a uniform field of stars. **(a)** Neglecting bending forces in the fabric makes the stars contract independently of each other, preventing the emergence of global curvature. **(b)** Using a flat rest shape for the rods makes the stars less curved than in reality. **(c, d)** Our complete model better reproduces the fabricated surface.

We use this expression to set the rest curvature of the rods about their width axis, $\bar{\kappa}_i^1$ in Equation 3.3. This model allows us to accurately reproduce the global curling up of a uniform star pattern into a rolled tube; see Figure 3.2. We provide in Figure 3.5 a comparison between our model and a model that ignores this bilayer effect, showing that our model better captures the behavior of a real-world fabricated surface.

3.2.4 Optimization

Defining the total energy $W(\mathbf{x}) = W_B(\mathbf{x}) + W_M(\mathbf{x}) + W_R(\mathbf{x}) + W_{connections}(\mathbf{x})$ where \mathbf{x} is a vector stacking all degrees of freedom, we compute the equilibrium shape of a star pattern by minimizing the total energy and finding $\mathbf{x} = \text{argmin } W(\mathbf{x})$ using the Newton-Raphson procedure with line search [90].

Starting from a flat initial guess \mathbf{x}_0 , the Newton-Raphson algorithm iteratively looks for \mathbf{x} such that $\nabla W(\mathbf{x}) = 0$ and $\nabla^2 W(\mathbf{x}) > 0$. To use this algorithm, we need to compute the first and second derivatives of W , which are implemented in our open-source library [61]. This algorithm iteratively solves a linear system involving the Hessian matrix $\nabla^2 W(\mathbf{x})$, this matrix may be non-positive, when that happens we regularize it by adding a multiple of the identity matrix. We report the runtime performances of this minimization procedure for all the examples shown in this chapter in Table 3.1.

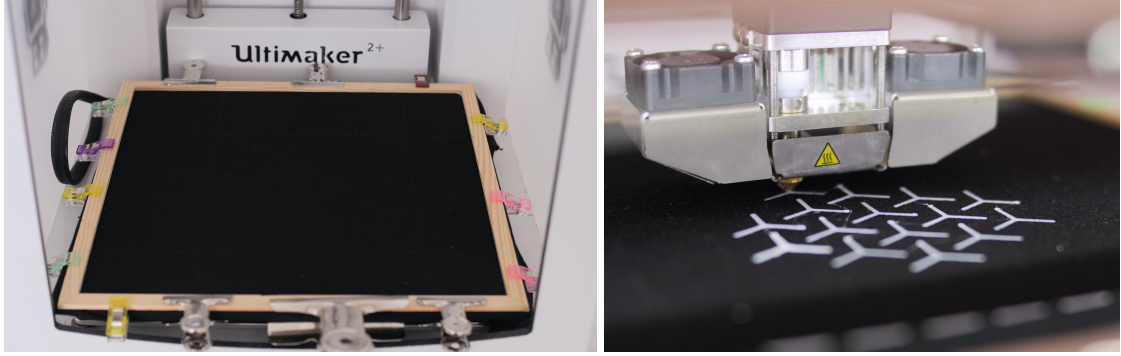


Fig. 3.6: Our 3D printing setup. We stretch the fabric and clamp it between a wooden frame and a plexiglas plate, which we attach to the build plate with magnets (left). The printer deposits plastic on the surface of the stretched fabric (right).

3.3 Fabrication

We now describe the hardware setup we used to produce our physical models, shown in Figure 3.6.

Our substrate is an elastic polyurethane fabric. We used a stretch factor of $s = 1.35$ (in both directions) in all of our experiments. To hold the fabric in place under tension, we built a custom, 26 by 26cm wooden frame and clamp the fabric between the frame and a plexiglass plate of similar dimensions (see Figure 3.6), which we fix to the print bed with magnets. To confirm that the fabric has been installed in its frame at the proper tension, we use a fabric marking pencil to draw a square with side length 10 cm near the center of the fabric before stretching it, and check that the square has side length $10s$ cm once the fabric is clamped in its frame.

In contrast to prior work [95] that used standard PLA printing filament, we performed our experiments using a more flexible TPU95A plastic to allow finer-grained control and higher range of star bending stiffness. We employ a UltiMaker2+ printer, which performs well with flexible filament. The printer must be calibrated to account for the thickness of the fabric and plate, effectively raising the height of the print bed. Indeed, we observed that without proper calibration to ensure that the nozzle begins directly above the fabric when extruding the first star layer, the melted plastic does not stick well to the fabric.

The material properties of printing filament can vary widely even between different spools of the same material purchased from the same manufacturer. To avoid relying on inaccurate material parameters from a material datasheet, we used Equation 3.17 to

directly calibrate the relative Young’s modulus of the plastic with respect to the fabric. We printed 4×10 strips of plastic of known thicknesses h_1 ranging from 0.3 mm to 0.45 mm over the fabric, and trimmed the fabric around each strip to produce a beam whose cross-section is a fabric-plastic bilayer. This beam flexes out of plane into a circular arc of approximately constant curvature; we photographed the side view (see Figure 3.4, left) and estimated the beam’s natural curvature κ from the dimensions of the bounding rectangle in the image plane. We then carefully removed the bottom fabric layer from the plastic rod and measured its rest curvature $\bar{\kappa}$ the same way (see Figure 3.4, right). Since we know the fabric’s thickness $h_2 = 0.8$ mm from measurements, the stretch factor $s = 1.35$, and the thickness h_1 of each printed plastic strip, we use linear regression and Equation 3.17 to deduce the ratio of Young’s moduli $E_1/E_2 = 3.2 \times 10^3$.

3.4 Evaluation

We first present several simulation experiments that illustrate how our model behaves with typical parameter settings. We then demonstrate the potential of our approach for architectural modeling by designing and fabricating a variety of freeform surfaces.

Effect of parameters. As described in section 3.2, the amount of contraction of the surface is influenced both by the arm length ℓ and thickness h of the stars. Figure 3.7 illustrates the effect of these two parameters when we vary them according to a linear radial gradient over a hexagonal domain. Setting ℓ smaller at the boundary of the domain and larger in the center gives more room for contraction at the boundary than in the middle, inducing the surface to buckle into a dome-like shape. A similar effect is achieved by using thinner stars on the boundary. In contrast, using long or thick stars at the boundary prevents contraction, so that the surface shape near the center is flatter. Using thick stars also results in smoother surfaces overall, while thin stars decorate the surface with multiple small bumps. While both thickness and arm length can be varied at the same time, we observed that ℓ usually has a greater effect on curvature, and so we used a fixed thickness of $h = 0.3$ mm for all other results. We also experimented with variations in arm width w , but this parameter had less impact than h on the surface shape — consistent with the fact that the bending stiffness of the star arms scales cubically in the arm thickness, but only linearly in its width.

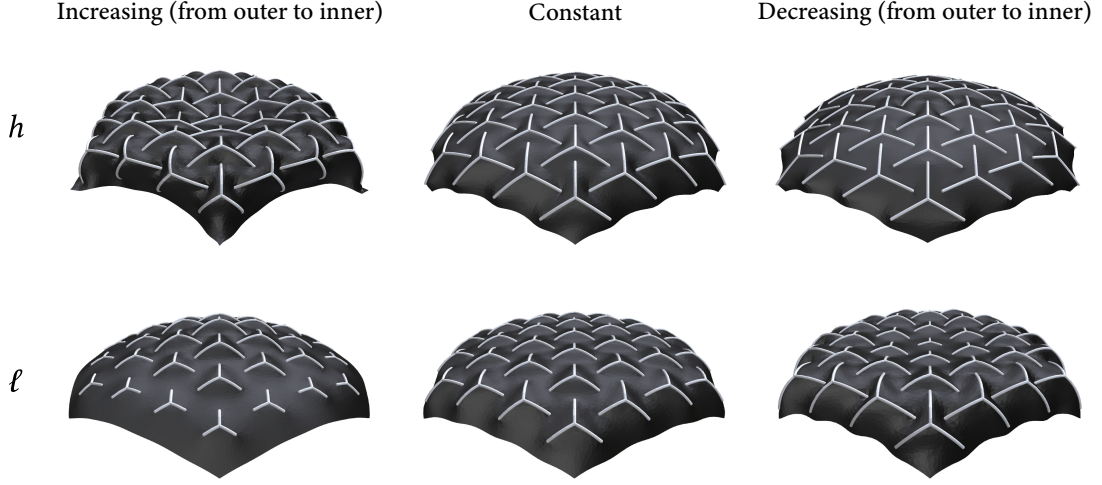


Fig. 3.7: We show the effect of two design parameters on the equilibrium shape of the star metamaterial: the length of the star arms ℓ and their thickness h . For this experiment we print 37 stars on a hexagon of fabric with side length 67 mm. In the top row we fixed $\ell = 10$ mm and set, from left to right, h with a decreasing radial gradient from 0.6 mm to 0.2 mm, a constant value of 0.4 mm and an increasing gradient from 0.2 to 0.6 mm. In the bottom row we fix $h = 0.3$ mm and prescribe three different radial gradients for ℓ : a decreasing gradient from 10 mm to 4 mm, and constant value of 7 mm and an increasing gradient from 4 mm to 10 mm.

Equipped with the ability to vary surface contraction locally, we can achieve different amounts of curvature by adjusting the rate of change of surface contraction. Figure 3.8 illustrates this control on a hexagonal domain fixed at its boundary, where we vary the size of the stars according to radial gradients of different profiles to produce surfaces ranging from a bell shape to a cone. For illustration purposes, we use a stronger stretching factor $s = 1.75$ in this experiment.

Example designs. Figure 3.9 showcases several architectural models that we created with our approach. For each result, we provide the input star pattern, the simulated surface, and a picture of the fabricated model. In all cases, we observe a close agreement between our simulation and reality.

The first two rows on top show singly-curved, cylinder-like shapes with first a curved cylindrical tunnel composed of a uniform field of stars which we fixed to the ground along two of its edges to prevent it to fold on itself, and second a cylindrical section with varying star sizes supported by six needles to model a small roof. The third row shows the result of a layout where large stars form a U-shaped boundary, while smaller stars contract

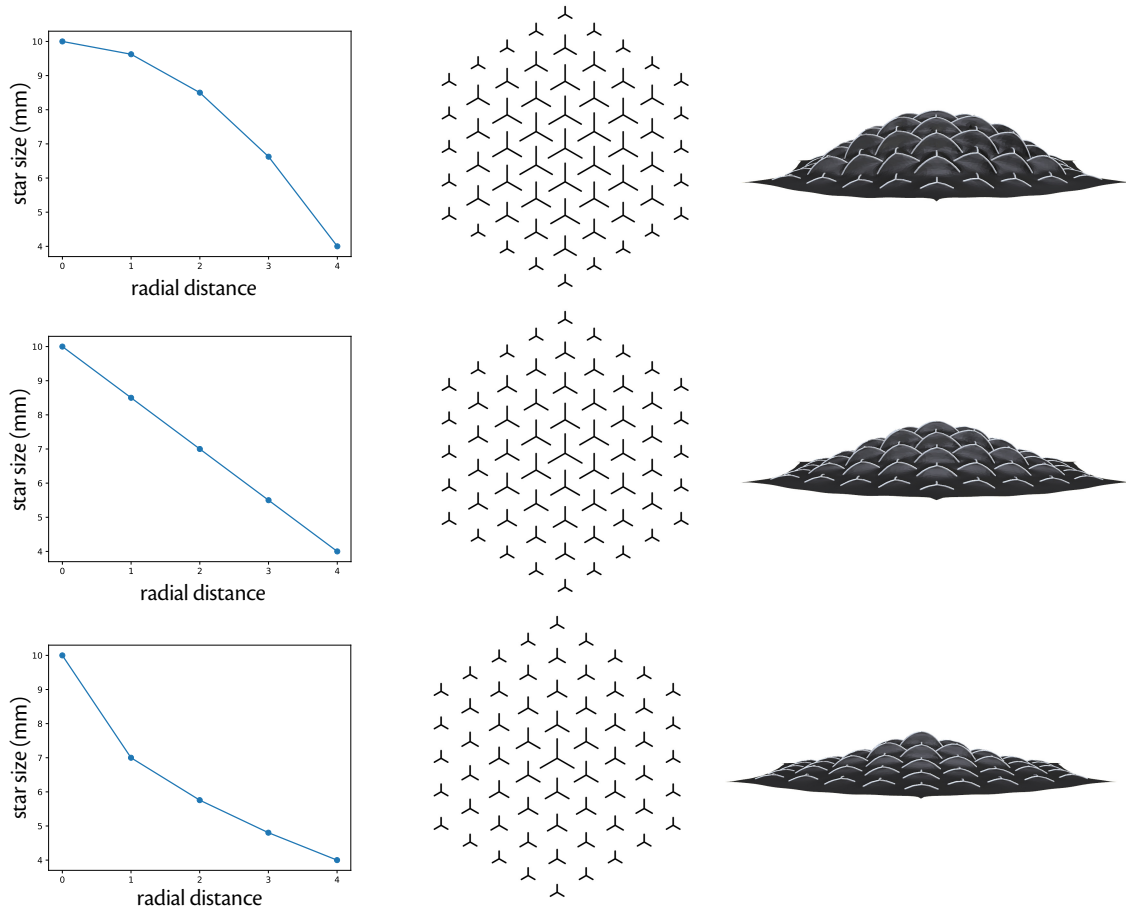


Fig. 3.8: Effect of different arm-length gradient profiles on the equilibrium shape of a hexagonal star pattern. Each row corresponds to a different function $\ell(x, y)$: on the left, a plot of ℓ as a function of distance to the hexagon center; in the middle, a diagram of the corresponding star pattern design; and on the right, the simulated equilibrium shape.

Name	Runtime	Nb of vertices
tunnel	23m57.8s	8020
roof	17m27.7s	7237
U shape	28m22.7s	10551
doubly-curved	40m0.4s	12140

Table 3.1: Runtime performance of the simulation on a PC with an Intel Xeon Gold 5118 processor running at 2.60GHz (single-threaded).

the surface in its center. The resulting surface takes the shape of an amphitheater. Only points along the U-shaped boundary were fixed. The last row shows a doubly-curved surface that we obtained using a hexagonal domain where we made the stars smaller in the middle and towards its three fixed corners. The three other corners are covered by bigger stars, which prevent them to contract as much as their surrounding, yielding buckling. We also applied a textile strengthener (Powertex) so that this surface supports its own weight.

Printing our models took 20 minutes on average, see Table 3.1 for simulation runtimes.

Discussion. The examples shown demonstrate that the specific pattern of disjointed stars we employed was well-suited to create positively curved surfaces and thus move away from the appearance of minimal surfaces that is prevalent in general tensioned structures. The tiling pattern of stars was flexible enough so that, by varying the size of the stars, we were able to create a variety of shapes, both singly- and doubly-curved. To aid in the design and shape exploration of this self-shaping textile medium, I developed a form-finding tool which allowed for easy visualization of the resulting shape. The form-finding method was inspired from existing work [95] but also modeled more complex behavior such as the bilayer effect of the plastic-fabric composite assembly. This form-finding method shows good agreement with the printed results, but the method may not be accurate for simulating some patterns different than tilings of stars. In the next chapter I will present a general-purpose simulation method for form-finding deployable, printed-on-fabric structures which is intended to work well on a variety of printed patterns.

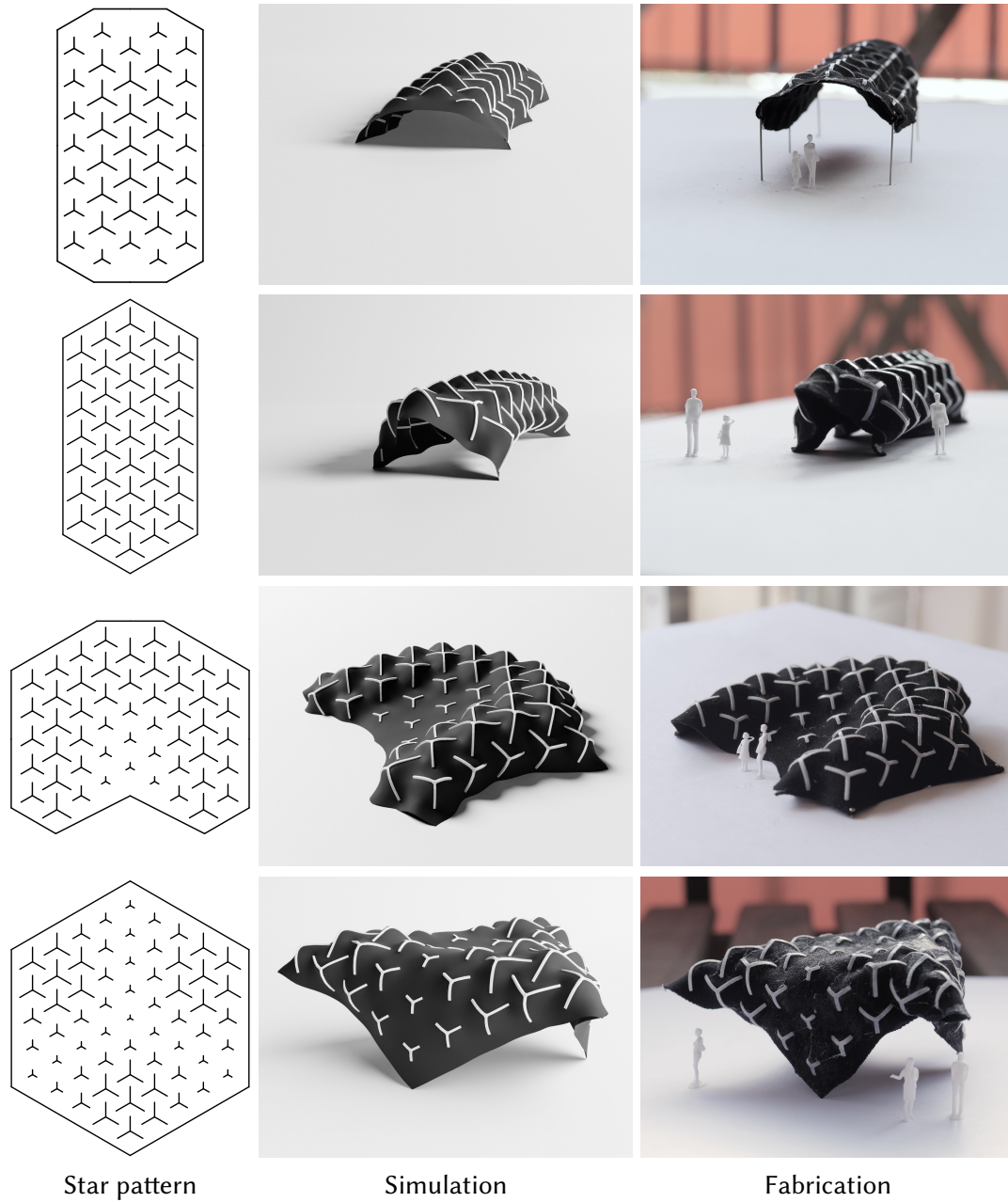
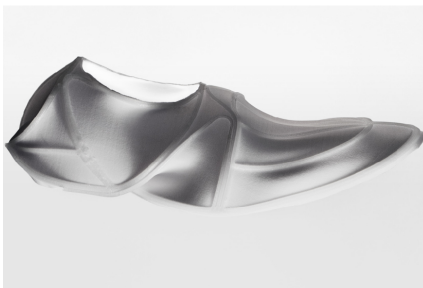


Fig. 3.9: Various architectural models created using our system, from top to bottom: a U-shape, a tunnel and a doubly-curved roof. For each result we provide the star layout to be printed (left), the simulated surface (middle), and the printed result (right).

BILAYER SHELL SIMULATION OF SELF-SHAPING TEXTILES

4.1 Introduction

Making simplifying assumptions about the behavior of materials is common practice when implementing simulation tools. Even the simple act of choosing a scale at which to observe a given physical phenomenon naturally leads to ignoring certain effects because they are not observable at the considered scale. These simplifying assumptions are often necessary as they allow to focus the computational effort on the phenomena which actually contribute to the results, but simplifications that are valid in one context might not be in another where they would introduce noticeable discrepancies between the actual object and its simulated counterpart. In the previous chapter, two of such simplifying assumptions were made, the fabric was thought to have an almost *isotropic* behavior and the rods' width was assumed to be negligible compared to their length. In this chapter, we are interested in developing a simulator that is not only capable of predicting the deployment of the shapes presented before, but also more general patterns of rod networks, general parallel curves, or even patterns which do not look like curves at all (see examples below). With that goal in mind, we can revisit those assumptions and see if they are still valid approximations.



Gubaran and Clopath [51]

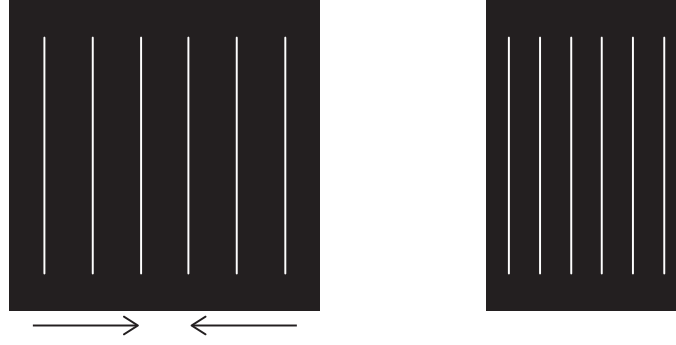


Erioli and Naldoni [37]

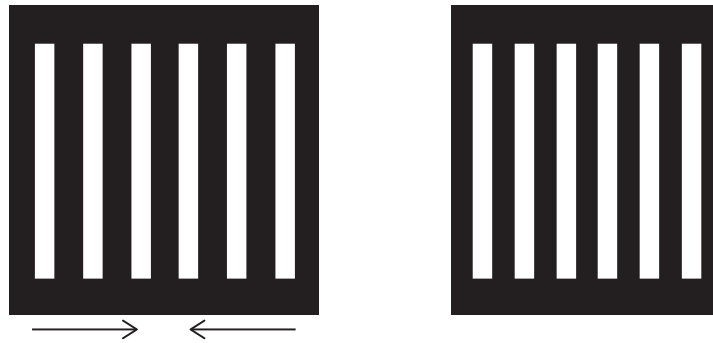


Fields [38]

Impact of width In the previous chapter – and similarly to Pérez et al. [95] – we used the Discrete Elastic Rods (DER) method to simulate the plastic curves. The DER method is based on a one-dimensional, reduced centerline representation which models the cross-section of the rods implicitly in the constitutive equations but does not explicitly model its geometry nor the impact it can have on the underlying substrate. This geometric representation proved to be effective in the case of sparse networks of connected curves [95] as well as in our case of 3-pointed stars. However, ignoring the width of curves can be detrimental when one wishes to modulate the retraction of the fabric in order to create some *metric distortion*. This can be easily seen by considering a simple pattern of parallel straight ribbons printed on a fabric that was stretched uniaxially along the x -axis. If the ribbons are modeled as thin curves, the fabric will be free to retract back to its original width l_0 :



However, if the ribbons have their surface properly modeled, the fabric will only be able to properly retract in the black regions where it is not impeded by the white plastic. The final width will therefore be larger than its initial width l_0 .



It is therefore necessary to have a proper *surface* model for the printed patterns so that the retraction of the fabric can be tracked accurately. A surface representation for the

plastic has also the advantage of being able to generalize to patterns which are not curves such as hexagon tilings [38].

Orthotropic behavior For the case of 3-pointed stars in the previous chapter, the fabric was modeled as an *isotropic* membrane which means that the simulated textile had the same properties in every direction. Therefore, the simulation was essentially rotation-independent: no matter how the stars were oriented, the results would be the same. And indeed we can observe experimentally that the approximation was valid in that



case: when printing individual stars or groups of stars in different orientations, the results are qualitatively the same (see inset). But this is not the case for all patterns, a pattern of parallel straight ribbons for example might give very different results depending on its orientation. I did an experiment where I printed two sets of parallel ribbon patterns with different thicknesses. The first set (left) was printed in one orientation, while the second (right) was printed at a 90° angle. The resulting shapes bent along two different axes:



A possible explanation for this experimental result would be that the samples have two different bending modes, one along the ribbons and one across, and which one dominates the other depends on the tensile forces applied by the surrounding fabric to the ribbons. In that case, the fabric having an *orthotropic* response – which means it has possibly different responses when rotated by a 90° angle – could explain this surprising result. Materials which have a one-axis reflection symmetry are known to exhibit orthotropic properties [115], and indeed looking at the structure of a typical elastic textile (which are

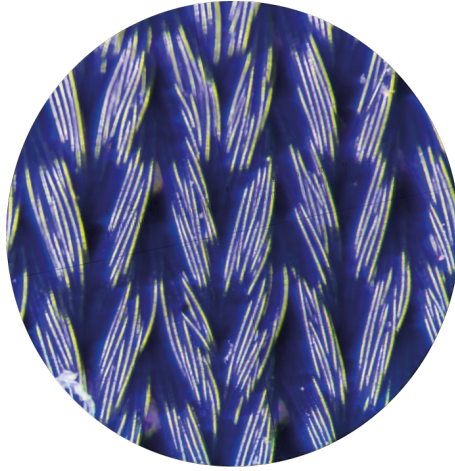


Fig. 4.1: Microscope view of a finely-knitted spandex textile (80 % polyamide, 20 % elastane).

usually knitted following specific stitch patterns for better compliance) we can see the knitting pattern exhibits such a reflection symmetry (see Fig. 4.1).

For a simulation method to work on as many printing-on-fabric scenarios as possible, it is therefore necessary to take into account both the exact width of the printed curves and the direction-dependent behavior of the stretched fabric. However, to the extent of my knowledge, there is no simulator for self-shaping textiles which takes into account both of these effects. As I mentioned before, both our method from Chapter 3 and the previous work of Pérez et al. [95] used a one-dimensional representation for the plastic curves coupled to an isotropic membrane model. The work of Stapleton et al. [132] who used the FEM software Abaqus to simulate a simple rectangular shape goes further: by modeling explicitly the surface of the plastic as well as the fabric, their model seems better suited to accurately track local metric variations induced by the width of the printed lines. However, they still modeled the fabric as an isotropic material which can be problematic, as highlighted above.

Overview. In this chapter, I will present preliminary work towards developing a general method to simulate the final deployed shape of printed-on-fabric patterns, by modeling both the finite width of the printed curves and the non-linear, orthotropic response of the fabric. The method is based on a physically accurate shell simulator which naturally takes into account the geometry of printed ribbons because of its surface representation, and is

tailored to reproducing bilayer effects such as the one encountered when attaching a rigid material to a pre-stretched substrate [147]. Instead of computing explicitly the curvature of the plastic-fabric bilayer and setting it as the rest curvature in a rod-based framework, our shell simulator directly integrates the bilayer effect within its constitutive equations. Therefore, this shell simulation method takes into account both intrinsic behaviors such as metric frustration due to the printed rods having finite width, and extrinsic behaviors such as bilayer effects due to the layers of plastic and fabric having different reference geometries.

To give accurate results, the simulator needs to be properly calibrated: the geometry of the final deployed shape can be very sensitive to changes in material properties such as Young's modulus of the different materials and in geometric properties such as the thickness of the different layers. We performed extensive measurements on two materials: an elastic textile material which can be used for printing-on-fabric on one hand, and a flexible filament used for 3D printing on the other hand. Using these measurements, I developed a data-driven material law which reproduces well the high-strain behavior of the stretched fabric when it is being released.

In summary, the two main contributions of this chapter are (1) a general-purpose simulation method able to accurately reproduce a variety of printed-on-fabric designs, and (2) a set of experimental results used to calibrate this simulation tool to particular materials.

This chapter is part of an ongoing collaboration between Victor Romero from Inria Grenoble, Etienne Vouga from UT Austin, and my supervisors Adrien Bousseau and Mélina Skouras. Victor Romero helped me perform the physical experiments described in the rest of this chapter.

4.2 Background

Bilayer shell simulation. Van Rees et al. [147] proposed a shell simulation method capable of modeling combinations of layers with different metrics by defining an *elastic energy inner product*. They showed how to compute the first and second rest fundamental forms of the bilayer as a function of the rest fundamental forms of the individual layers. Chen et al. [23] used this method to simulate environmental effects such as moisture or temperature gradients. We adopt this framework to model the bilayer formed by the combination of plastic and fabric whose respective first fundamental forms and material

properties can be combined into a reduced shell representation (see Section 4.3). Even though this model was based on a small-strain analysis, we show how to adapt it in our case where the fabric is stretched well past its linear regime.

Fabric simulation. Computer graphics has had a long history of trying to simulate textiles, notably because of the need to realistically animate the clothes of virtual characters. In that context, the deformations are mostly restricted to the small-strain regime and approximate models such as the popular approach of Baraff and Witkin [10] were shown to work well in many cases.

However, the textiles are subject to extreme deformation in our case, for the simulation to match real-life experiments we need to accurately reproduce the full stress-strain curve and not just its derivative at 0. Unfortunately, it is impossible to derive a simple stress-strain relationship for say, knitted fabrics, because fabric are heterogeneous materials made from a multitude of fibers arranged into threads. Therefore it is difficult to describe the material response of textiles in terms of continuum mechanics, as we can do for e.g. rubbers with the neo-Hookean or Mooney-Rivlin models. Instead, there are two popular options: either simulate the fabric at a smaller, micro- or meso-scale, or use a data-driven material model optimized for a given fabric material. Simulating at smaller scales is the approach of yarn-based methods [29], the idea is to simulate the textile at the level of individual threads where each thread is modeled as an elastic rod [14] and contact and friction between threads is tracked as well. Unfortunately, this type of method can become prohibitively expensive when considering a large area of fabric because of the number of contact points involved. Homogenization-based methods can mitigate the issue by combining the level of detail of yarn-based methods with the speed of shell simulations [130] but they still need proper calibration and measurements to be physically accurate.

Another approach is to directly measure the elastic response of the material and derive a data-driven material model from those measurements. A way to do it is by measuring so-called *stress-strain* curves, expressing the 2nd Piola-Kirchhoff stress σ as a nonlinear function of the Green strain $\epsilon = \frac{1}{2}(F^T F - I)$, where F is the deformation gradient. Volino et al. [151] showed that we can express the entries of the stress matrix σ_{ij} as a nonlinear function of the strains ϵ_{ij} to derive an orthotropic data-driven material model. We take inspiration from their method, even though they do not take into account transverse

effects, i.e. couplings between strains in both directions due to the Poisson effect.

4.3 Material model

Our printed-on-fabric metamaterial is modeled as an inhomogeneous shell which is comprised of two different materials: a *fabric* material and a *bilayer* material, i.e. each point on the surface is either in a fabric-only area, or a bilayer area (see Fig. 4.2). Both materials are modeled as thin shells but with different material models: the bilayer shell model is based on the work of van Rees et al. [147] which models the curvature of bilayers by looking at metric differences between the two layers; and the fabric is modeled using a custom data-driven material model which is fitted against the experimental data described in section 4.4.

Bilayer material Similar to the previous chapter (Eq. 3.9), the elastic energy of a bilayer composed of two materials with rest first fundamental forms a_{r1} and a_{r2} , Young's

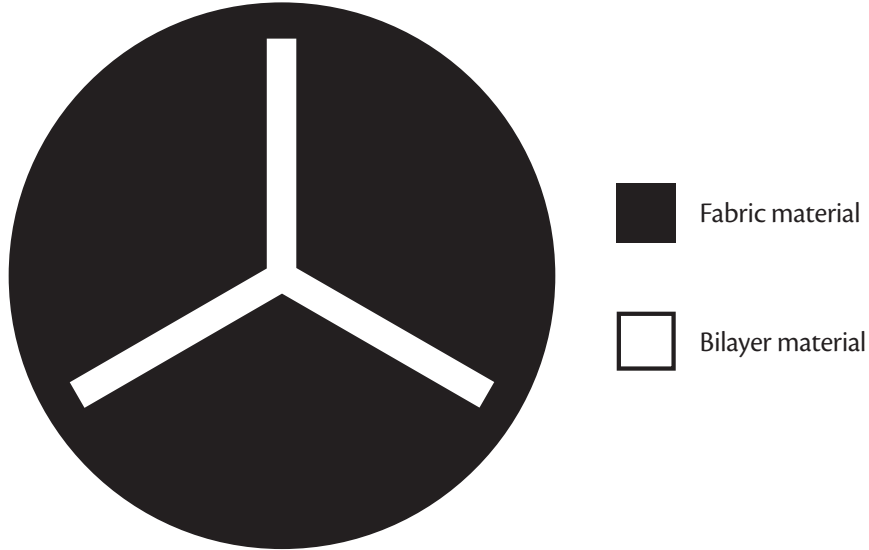


Fig. 4.2: Example of modeling a 3-pointed star shape similar to the ones in Chapter 3: the black area corresponds to the fabric and is modeled using a custom data-driven material model, the white area corresponds to a plastic-fabric bilayer modeled using the method of van Rees et al. [147].

moduli E_1 and E_2 , and thicknesses h_1, h_2 is defined as:

$$\begin{aligned}
 W_{BL} = & \frac{1}{2} \int_U E_1 \left[\frac{h_1}{8} \|a_{r_1}^{-1} a_c - I\|^2 + \frac{h_1^3}{24} \|a_{r_1}^{-1} b_c\|^2 \right. \\
 & \left. + \frac{h_1^2}{8} \langle a_{r_1}^{-1} a_c - I, a_{r_1}^{-1} b_c \rangle \right] \sqrt{\det a_{r_1}} \, dx \, dy \\
 & + \frac{1}{2} \int_U E_2 \left[\frac{h_2}{8} \|a_{r_2}^{-1} a_c - I\|^2 + \frac{h_2^3}{24} \|a_{r_2}^{-1} b_c\|^2 \right. \\
 & \left. - \frac{h_2^2}{8} \langle a_{r_2}^{-1} a_c - I, a_{r_2}^{-1} b_c \rangle \right] \sqrt{\det a_{r_2}} \, dx \, dy
 \end{aligned} \tag{4.1}$$

where $\langle A, B \rangle = \frac{\nu}{1-\nu^2} \text{tr}(A) \text{tr}(B) + \frac{1}{1+\nu} \text{tr}(AB)$ is the elastic inner product associated to a material with Poisson's ratio ν , and $\|A\|^2 = \langle A, A \rangle = \frac{\nu}{1-\nu^2} \text{tr}^2(A) + \frac{1}{1+\nu} \text{tr}(A^2)$ is the elastic energy norm.

a_{r_1} and a_{r_2} are respectively the first fundamental forms of the fabric and plastic layers. Depending on how the fabric was stretched, a_{r_1} will have different entries. Two use cases are common when fabricating self-shaping textiles, the first case is *uniaxial strain*, meaning the fabric was stretched only in one direction, the second case is *isotropic stretch* which means the fabric was stretched equally in both directions. If we call s the ratio of lengths between the undeformed and the deformed state, we will have $a_{r_1} = \begin{pmatrix} \frac{1}{s^2} & 0 \\ 0 & 1 \end{pmatrix}$ for uniaxial strain and $a_{r_1} = \frac{1}{s^2} I$ for isotropic strain. Since the plastic is initially strain-free, its first fundamental form corresponds to the identity matrix: $a_{r_2} = I$

a_c and b_c are respectively the first and second fundamental forms of the *current configuration*. We discretize Equation 4.1 so that they are constant on each triangle. Following Chen et al. [23], we compute a_c on the triangle formed by vertices v_i, v_j, v_k as:

$$a_c = \begin{pmatrix} \|v_j - v_i\|^2 & (v_j - v_i) \cdot (v_k - v_i) \\ (v_j - v_i) \cdot (v_k - v_i) & \|v_k - v_i\|^2 \end{pmatrix}$$

b_c is computed with the “triangle with flaps” stencil of Grinspun et al. [46]:

$$b_c = \begin{pmatrix} (n_j - n_i) \cdot (v_j - v_i) & (n_j - n_i) \cdot (v_k - v_i) \\ (n_k - n_i) \cdot (v_j - v_i) & (n_k - n_i) \cdot (v_k - v_i) \end{pmatrix}$$

n_i, n_j, n_k are normals defined on the edge opposite v_i, v_j, v_k respectively, and computed by averaging the normals of their adjacent faces.

Equation 4.1 gives the elastic energy of a bilayer composed of homogeneous materials with the same Poisson's ratio ν . In our case, we use the average value of the Poisson's ratios of the two layers: the fabric's Poisson's ratio is measured from videos of a uniaxial tensile test as the ratio of transverse contraction to axial strain and is estimated to be about 0.3; the plastic material used (thermoplastic polyurethane) is known to be almost incompressible [103] which means a value of 0.5.

Moreover, the fabric layer is not isotropic, we will see in section 4.4 that we can compute its Young's modulus E_1 by averaging the tangent of the stress-strain curve upon unloading over a sampling of different orientations.

Fabric material model As is standard for modelling cloth [84, 154], we separate the bending and membrane contributions of the energy:

$$W_F = W_{\text{bending}} + W_{\text{membrane}}.$$

For the bending contribution, as in the previous chapter (Eq. 3.1) we use the Discrete Shell model of Grinspun et al. [47]:

$$W_{\text{bending}} = k_B \sum_i \frac{3\|\bar{\mathbf{e}}_i\|^2}{\bar{A}_i} \left(2 \tan\left(\frac{\theta_i}{2}\right) \right)^2$$

For the membrane energy, since textiles are inhomogeneous materials made from a multitude of fibers, it is difficult to describe their material response with a simple law derived from continuum mechanics as is possible for e.g. rubbers with the neo-Hookean or Mooney-Rivlin models. Instead, we define a custom parametric material model designed to reproduce well the nonlinear response of the fabric especially when released after a given amount of pre-stretch. For each triangle we can write the elastic energy as a function of its Green strain tensor written in Voigt notation as $\boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_{11} & \varepsilon_{22} & 2\varepsilon_{12} \end{pmatrix}^T$:

$$\Psi(\boldsymbol{\varepsilon}) = \frac{1}{1-\nu^2} \boldsymbol{\varepsilon}^T \begin{pmatrix} \alpha_1 & \sqrt{\alpha_1 \alpha_2} \nu & 0 \\ \sqrt{\alpha_1 \alpha_2} \nu & \alpha_2 & 0 \\ 0 & 0 & \alpha_3(1-\nu^2) \end{pmatrix} \boldsymbol{\varepsilon} - \varphi_1(\varepsilon_{11}) - \varphi_2(\varepsilon_{22}) \quad (4.2)$$

where φ_1 and φ_2 are symmetric log-barrier functions:

$$\begin{aligned} \varphi_i(\varepsilon_{ii}) &= \beta_i \left((\varepsilon_{ii} - \gamma_i) \log\left(\frac{\gamma_i - \varepsilon_{ii}}{\gamma_i}\right) - \varepsilon_{ii} \right) \text{ if } \varepsilon_{ii} > 0, \\ &= -\beta_i \left((\varepsilon_{ii} + \gamma_i) \log\left(\frac{\gamma_i + \varepsilon_{ii}}{\gamma_i}\right) - \varepsilon_{ii} \right) \text{ if } \varepsilon_{ii} < 0. \end{aligned} \quad (4.3)$$

and

$$W_{\text{membrane}} = h \sum_i A_i \Psi(\epsilon_i).$$

In the small strain regime, $\varphi_i \approx 0$ and the model has the form of an orthotropic StVK energy [78], it then increases exponentially in the high strain regime to match the measured stress-strain curves in section 4.4. Since the function is symmetric with respect of the y -axis, it also penalizes exponentially negative strains which is useful to prevent triangle inversions. The model is parameterized by seven coefficients: $\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \gamma_1, \gamma_2$ which are optimized to match the experimentally measured response, and a Poisson's ratio ν measured experimentally.

4.4 Measurements and fitting

In this section, I will describe the different experiments we conducted and how we used these measurements to calibrate the simulation model presented above. We considered two materials: an elastic textile which is representative of the kinds of materials used for 3D printing on stretched fabric, and a flexible 3D printing plastic filament. The textile is a finely-knitted lycra fabric composed of 80 % polyamide and 20 % elastane, while the printing filament material is a thermoplastic polyurethane known as Ultimaker TPU 95A. We conducted three types of measurements: uniaxial stretch tests (performed on both the fabric and plastic material), shear tests and cantilever tests (performed on the textile only).

4.4.1 Uniaxial stretch

For this test, samples were prepared in a standard *dogbone* shape, the plastic was 3D printed into a 15cm long by 1cm wide shape, while several fabric samples were laser cut out of a 11cm long by 2cm wide template. The fabric samples were cut out at different orientations to measure the orthotropy of the material, we measured the tensile response of samples oriented at 0, 15, 30, 45, 60, 75, and 90° respectively (measured from the horizontal axis in Fig. 4.1).

The tensile measurements were made using an Instron 5865 machine with a 50 N force sensor which tracks both the displacement d of the clamped endpoints and the force applied f (called the *response* of the material). These force-displacement curves then have

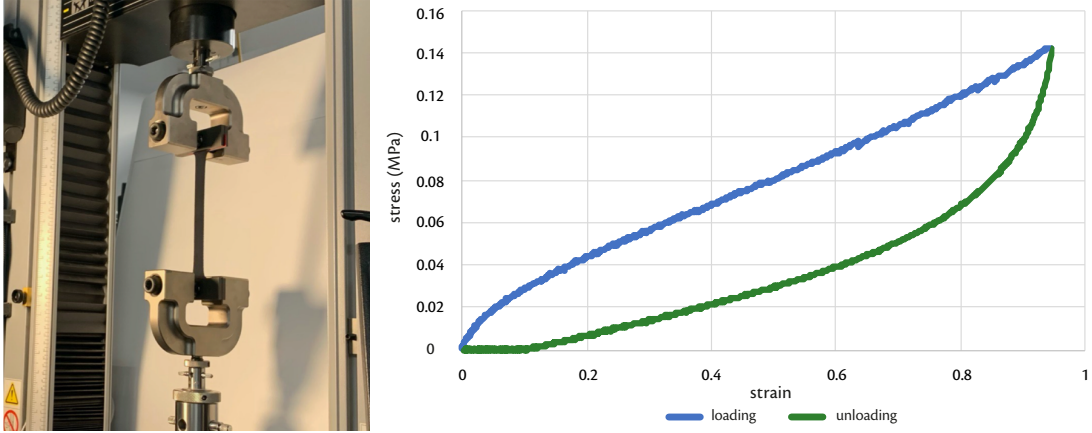


Fig. 4.3: *left*: uniaxial stretch testing setup, *right*: the stress-strain curve of a fabric sample forms a cycle because the path followed is not the same between loading and unloading.

to be converted to *stress-strain* curves in order to be able to extract useful calibration information such as the Young's modulus of the materials. We convert these values into entries of the second Piola-Kirchhoff stress tensor $\sigma = \frac{\partial \Psi}{\partial \epsilon}$ and the Green strain tensor ϵ which are accurate quantities for large strains. For uniaxial stretch, the diagonal entries of these 2×2 matrices can be computed from the force-displacement curves of either the 0° or 90° samples:

$$\begin{aligned} \epsilon_{ii} &= \frac{d}{L} + \frac{d^2}{2L^2} \\ \sigma_{ii} &= \frac{1}{w} \left(f \frac{L}{d + L} \right) \end{aligned} \quad (4.4)$$

the $i = 1$ values can be computed from the 0° sample and the $i = 2$ values can be computed from the 90° one.

Fig. 4.3 shows a picture of the rig setup. The testing machine stretched and released the samples by performing load-unload cycles, which created loops characteristic of a hysteresis behavior when plotting the stress-strain curves. This hysteresis, or path-dependent behavior, is likely caused by the internal friction between fibers of the fabric which rearrange as the textile gets stretched. Since we want to model the behavior of the textile once it has been stretched and gets released, we are really interested in the *unloading* part of the curve (in green in Fig. 4.3).

We first use the stress-strain curves of the plastic and the different fabric samples to calibrate the bilayer material model (Equation 4.1), and in particular, to measure the

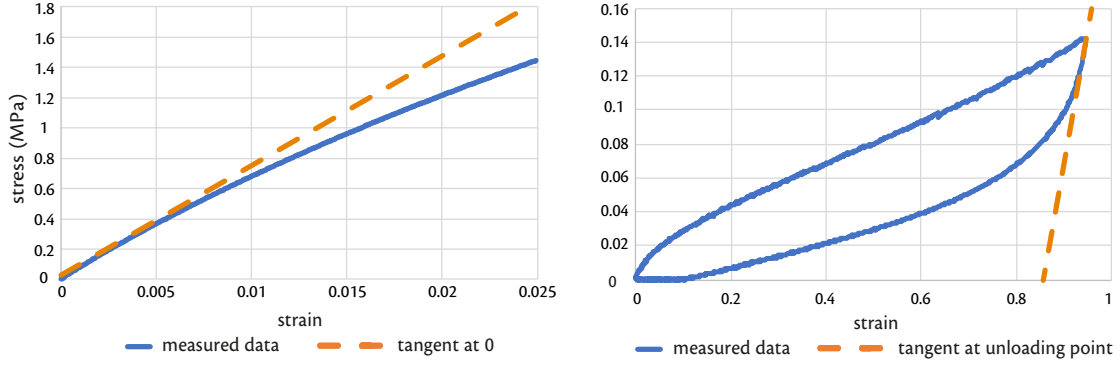


Fig. 4.4: *left*: stress-strain curve for the plastic material, the slope of its tangent at 0 is the material's Young's modulus; *right*: stress-strain curve for the fabric material, here we are interested in the slope of the tangent upon unloading.

Young's moduli E_1 and E_2 of the two layers. Classically, the Young's modulus is the slope of the tangent at 0 of the stress-strain curve, it is a linear approximation of the stress-strain behavior for small strains, we can therefore compute the plastic's Young's modulus E_1 by finding the tangent at zero of its stress-strain curve (Fig. 4.4, left). For the fabric substrate however the situation is a bit different, since it was pre-stretched when the plastic-fabric bilayer was formed, it does not make much sense to consider an approximation of its response for strains close to zero since the plastic layer prevents it from ever going back to its original length. Instead, the linear approximation should start from the maximum strain value, E_2 is therefore computed as a pseudo "Young's modulus" by finding the slope of the tangent at the point of release (Fig. 4.4, right). Since this value is measured at the point of unloading, it may depend on how far the sample was stretched, to account for a dependency on the initial strain of the fabric, we performed two tests in which the fabric samples were stretched up to 45 % and 70 % of their initial lengths, and linearly interpolate between the two measured values to simulate self-shaping textiles pre-stretched with a different strain value.

We also use the uniaxial stretch data to find the coefficients of the fabric material model. We can compute the stress tensor by differentiating Equation 4.2 (assuming $\varepsilon_{ii} > 0$):

$$\sigma = \frac{\partial \Psi}{\partial \varepsilon} = \frac{1}{1 - \nu^2} \begin{pmatrix} \alpha_1 & \sqrt{\alpha_1 \alpha_2} \nu & 0 \\ \sqrt{\alpha_1 \alpha_2} \nu & \alpha_2 & 0 \\ 0 & 0 & \alpha_3 (1 - \nu^2) \end{pmatrix} \begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{pmatrix} - \begin{pmatrix} \beta_1 \log \left(\frac{\varepsilon_{11} - \gamma_1}{\gamma_1} \right) \\ \beta_2 \log \left(\frac{\varepsilon_{22} - \gamma_2}{\gamma_2} \right) \\ 0 \end{pmatrix}. \quad (4.5)$$

We can use the relationship $\nu = -\frac{\varepsilon_{22}}{\varepsilon_{11}}$ to express σ_{11} as function of ε_{11} , and σ_{22} as function

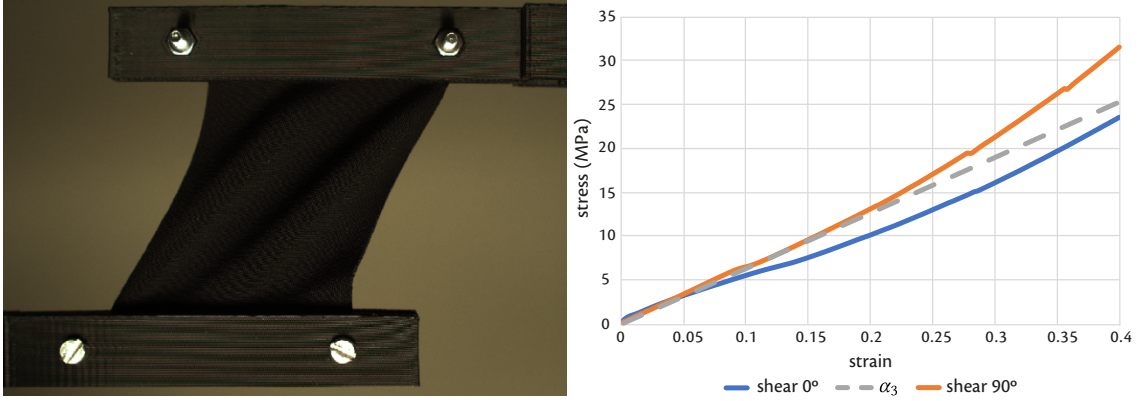


Fig. 4.5: *left*: shear testing setup *right*: stress-strain curves for a square sample oriented at 0° and 90° , and α_3 as a linear approximation of the average of both curves.

of ε_{22} :

$$\sigma_{11} = \frac{1}{1-\nu^2} \left(\alpha_1 - \sqrt{\alpha_1 \alpha_2} \nu^2 \right) \varepsilon_{11} - \beta_1 \log \left(\frac{\varepsilon_{11} - \gamma_1}{\gamma_1} \right) \quad (4.6)$$

$$\sigma_{22} = \frac{1}{1-\nu^2} \left(\alpha_2 - \sqrt{\alpha_1 \alpha_2} \nu^2 \right) \varepsilon_{22} - \beta_2 \log \left(\frac{\varepsilon_{22} - \gamma_2}{\gamma_2} \right). \quad (4.7)$$

We then fit a function of the form $\sigma_{ii}(\varepsilon) = a\varepsilon_{ii} - b \log \left(\frac{\varepsilon_{ii} - c}{c} \right)$ which allows to directly retrieve β_1 , β_2 , γ_1 and γ_2 . For α_1 and α_2 , solving a simple nonlinear system allows to compute them as a function of ν . These parameters also depend on how much was the fabric stretched, we can also linearly interpolate the fitted values for different amounts of stretch as was done for E_2 .

4.4.2 Shear tests

The last coefficient needed for the membrane parametric model is α_3 which corresponds to the shear modulus. To measure this coefficient we clamp two parallel edges of a square sample and move one of its edges laterally while probing the response (Fig. 4.5, left). We performed the measurement twice on the same sample in two different orientations, 90° from each other. The results at 0° and 90° showed an approximately linear response which corresponds well with the choice of expressing σ_{12} as a linear function of ε_{12} in the parametric model. The two shearing experiments correspond to the same strain value ε_{12} and therefore should give the same stress σ_{12} as a result. However, a surprising experimental result was that the response was not the same at a 0° and 90° orientation (Fig. 4.5, right), this could either be due to errors in the experimental process or to more

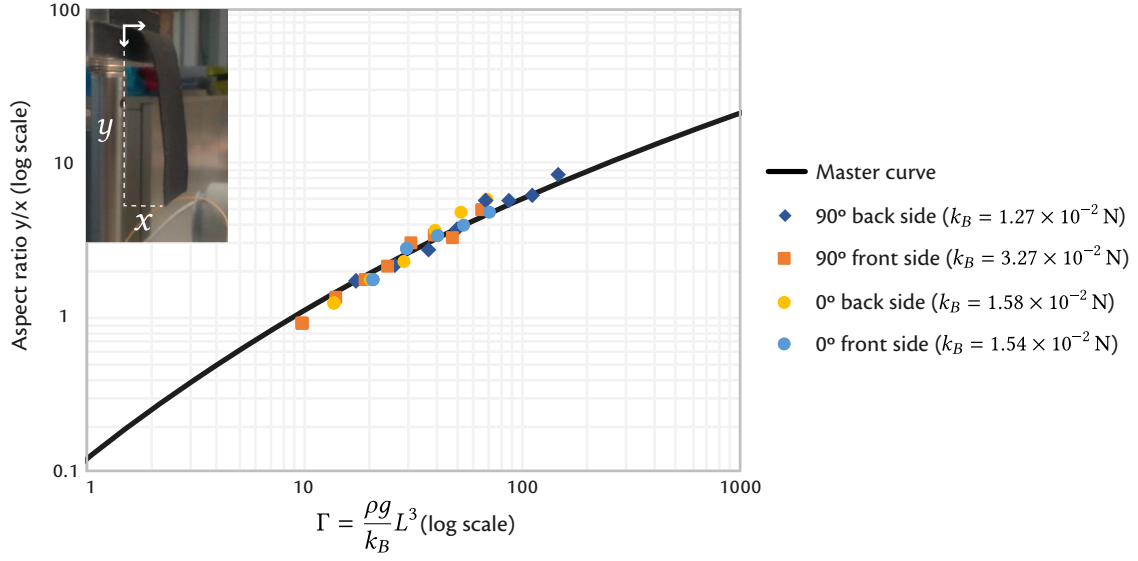


Fig. 4.6: Matching measurements of cantilevered shapes against the “master curve” of Romero et al. [108].

complex effects occurring within the fabric which our model cannot reproduce. Another possibility is that the curves are diverging after the small strain regime because of the orthotropy of the material: at large strains, the deformation is not pure shearing anymore but a mix of shearing and stretching, and since the stretching response is direction dependent this could explain the deviation between the curves. Since our model cannot explain this divergence between the two curves, we compute α_3 as the average linear regression of each curve.

4.4.3 Bending tests

The last coefficient needed for the fabric’s material model is the *flexural coefficient* k_B of the bending energy. This coefficient is usually expressed as a function the Young’s modulus E , the thickness h and the Poisson’s ratio ν for a homogenous material:

$$k_B = \frac{Eh^2}{12(1 - \nu^2)}$$

but we can directly measure it by performing a cantilever test in which a fabric ribbon sample is clamped horizontally and a length L of itself is submitted to gravity. Romero et al. [108] show the relationship between a unitless *gravito-bending* parameter $\Gamma = \frac{\rho g}{k_B} L^3$ – where ρ is the mass density and g is the acceleration of gravity – and the aspect ratio y/x

E_1 (MPa)	E_2 (MPa)	α_1	α_2	α_3	β_1	β_2	γ_1	γ_2	k_B (N)
72.3	1.05	24161	90890	29676	20512	0.9626	0.9637	63419	0.0192

Table 4.1: Parameters measured on a TPU95A thermoplastic filament and a spandex textile stretched up to 70 % of its initial length.

of the bounding rectangle of the cantilevered sample (see Fig. 4.6, inset). This relationship is universal in the sense that the pair $(\Gamma, y/x)$ will always be on a specific curve, called the *master curve*, no matter what the material properties of the sample.

To find the value of k_B , we therefore measure y/x for different values of L and find the coefficient k_B such that the different points $(\frac{\rho g}{k_B} L^3, y/x)$ are as close as possible to the curve in the least squares sense. We performed the test on two different orientations of the fabric (0° and 90°) and tested them both front side up and back side up, for a total of 4 different experiments. The results (Fig. 4.6) show a difference in k_B between the 0° and 90° orientation, which is not surprising given the structure of the knitted textile (Fig. 4.1). The difference between bending front side up and back side up was found to be negligible in the 0° case – meaning the resistance to bending is essentially symmetric in that direction – but for the 90° case the flexural coefficient is almost 3 times bigger on one side compared to the other, which can be explained by the fact that knitted textiles in general do not exhibit mirror symmetry between their front and back sides and therefore can have fairly different responses between bending upwards and downwards. The Discrete Shells model that we use to model bending does not account for these direction-dependent effects and is only weighted by one flexural coefficient (k_B), therefore we compute k_B as mean value between the 4 measured ones. All the different parameters used in the bilayer model (E_1, E_2), the membrane energy ($\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \gamma_1, \gamma_2$) and the bending energy (k_B) are summed up in table 4.1.

4.5 Preliminary results and validation

To validate our simulation method, I performed a series of basic tests, comparing the simulation results against experimental data and simulating simple shapes. To check that the fabric’s material model reproduces well the measured behavior, I ran several simulations in which a rectangle was uniaxially stretched. Depending on how the rectangle was oriented in global coordinates, the integrated forces along each edge were different and matched well with the experimental data (Fig. 4.7).

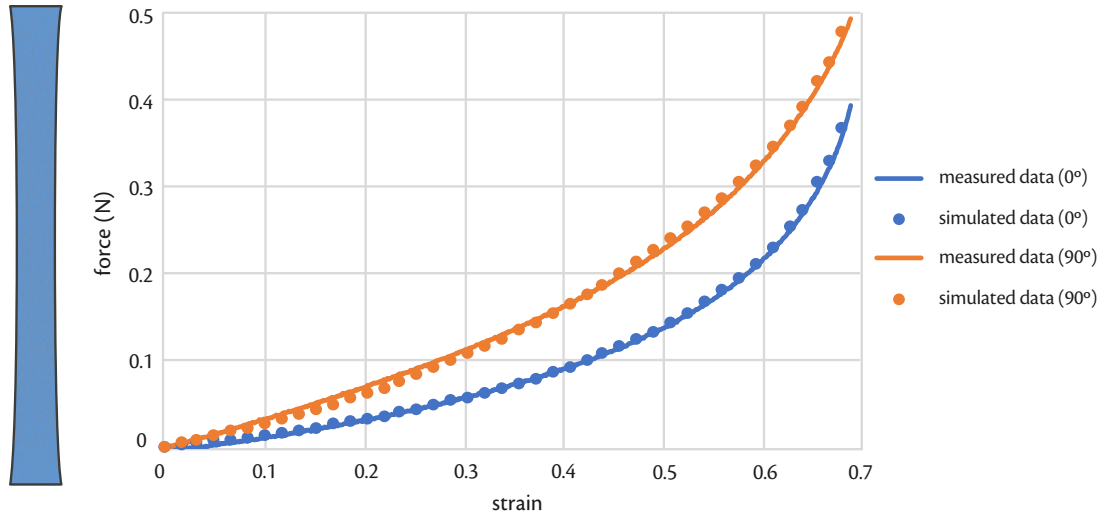


Fig. 4.7: *left*: simulated sample undergoing uniaxial strain, *right* comparison between measured force values and simulated ones for different orientations.

I also simulated a simple pattern of parallel ribbons of the same thickness. The bilayer effect makes the assembly roll into a cylinder as expected from similar printed results in Chapter 5 (Fig. 4.8).



Fig. 4.8: Simulation of a pattern of parallel ribbons rolling into a cylinder.

4.6 Discussion and limitations

The goal of this chapter was twofold: better understanding, by means of physical experiments, the mechanics of pre-stretched textiles when released and their interaction with thermoplastics printed on top, and deriving a simulation model from those measurements and experiments which should be both accurate and efficient.

At the moment, this method is still very much work in progress and needs to be validated against a variety of printed patterns to assess its accuracy. To check that the bending behavior caused by the bilayer effect is modeled correctly, we will compare the curvature of simulations of parallel ribbon patterns such as the one shown in Fig. 4.8 with the measured curvature of their fabricated counterparts. To check that the intrinsic contraction behaves similarly to printed patterns, we can print a pattern from next chapter which reproduces half a torus by having a lower density of curves on the inside of the flattened shape and a higher density of curves on the outside (Fig. 5.2). After performing these validations we will try to simulate our own printed patterns and a variety of other curve layouts inspired from previous work (such as the self-shaping textiles shown in introduction).

However, we mentioned in chapter 3 that the printed plastic material exhibited a certain amount of plasticity (Fig. 3.4) which has not yet been taken into account in a systematic way in the model, and therefore might cause the simulated results to be stiffer than the printed ones, this could prevent the method to accurately reproduce extrinsic curvature in particular. Another issue with the method is the amount of experimental data it requires to find the parameters for given plastic and fabric materials, it would be useful to have a way to easily calibrate the simulation method to other unseen materials (provided they are not too different). One way to do this would be to propose a set of tests that would each be used to measure a specific parameter. For example, measuring the curvature of an individual bilayer ribbon can be used to derive the Young's moduli ratio E_1/E_2 or measuring a series of parallel ribbons at different orientations gives information about the orthotropic fabric response.

INVERSE DESIGN OF SELF-SHAPING TEXTILES

5.1 Introduction

The previous chapters were about the form-finding of self-shaping textiles created by 3D printing on stretched fabric: given a specific pattern to be printed, the goal of the form-finding is to find a deployed shape which satisfies an equilibrium of forces between the pre-stretched fabric and the more rigid plastic printed on top. In this chapter we tackle the *inverse problem* instead: given a target surface that we want to reproduce, the goal is to find an optimal layout to be printed so that the deployed shape is as close as possible to the input. This is a difficult problem as it amounts to finding a mapping between a 3D surface and a flattened one which preserves the dimensions of all the printed shapes, since the plastic is essentially incompressible during deployment.

To solve this problem, we focus on printing patterns of almost-parallel ribbons on both sides of a pre-stretched elastic textile. This pattern allows to control both the metric by modulating the density of printed ribbons, and extrinsic curvature via the bilayer effect (Figure 5.1). We examine the interplay of geometry and physics that governs the

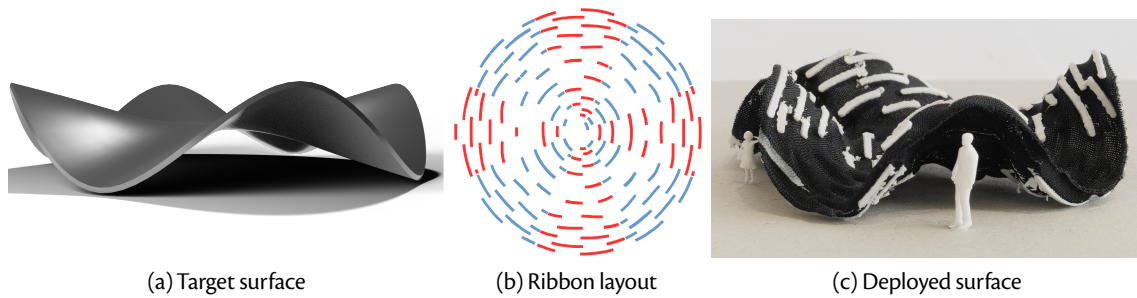
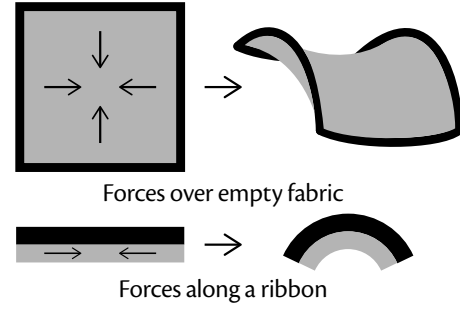


Fig. 5.1: Given a target 3D surface (a), our method computes a flat layout of plastic ribbons (b) which, when printed on pre-stretched fabric, causes the surface to deploy into 3D when the fabric is released (c). We designed a custom frame to easily print ribbons on both sides of the fabric (b, red and blue), allowing us to reproduce surfaces with positive and negative extrinsic curvature.

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 [51, 95]. 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 [95] (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*

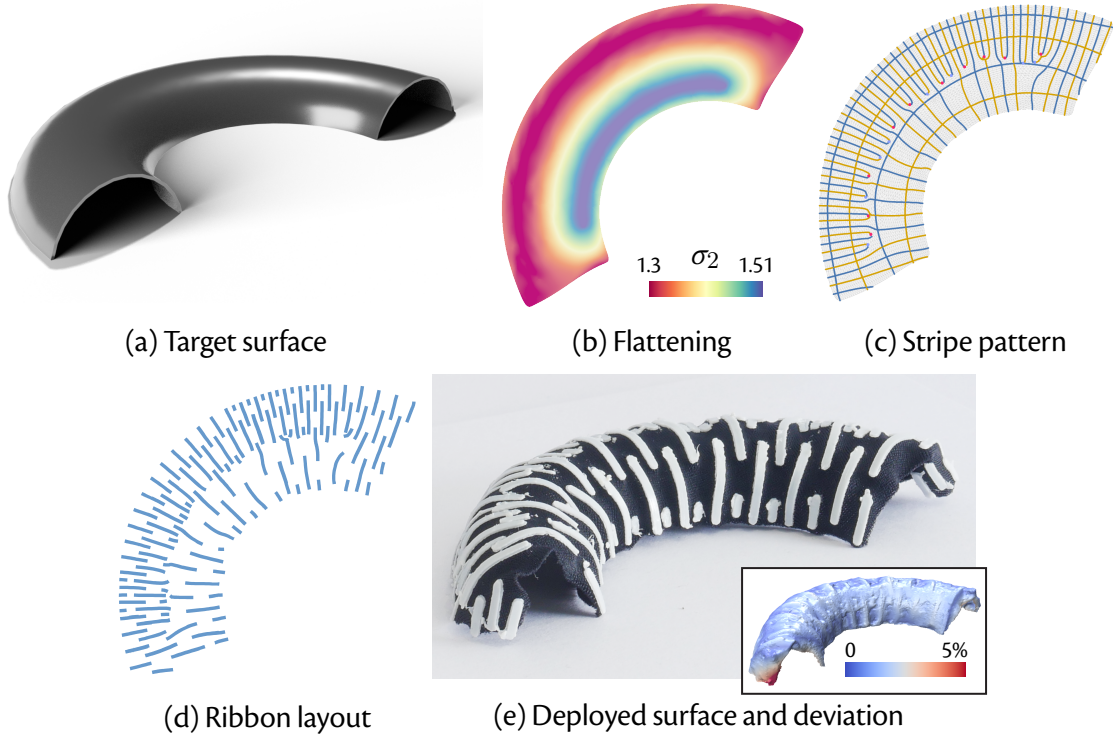


Fig. 5.2: Overview of our method. Given a target surface (a), we first compute its principal curvatures, along which plastic ribbons will be placed. We compute a flattening of the 3D surface into the plane, with bounded stretching (b; colors indicate the magnitude of stretching σ_2 in the ribbon direction). 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).

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 [24, 68], 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 5.2). 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 find the optimal ribbon thickness to obtain the extrinsic curvature magnitude.

In contrast to the direct design tools presented in the previous chapters, the inverse design method presented in this chapter is based on geometric principles rather than physical simulation: 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 [79]. While recent methods relied on a similar procedure to bound distortion for other fabrication techniques [3, 93], 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 directional curvature. We generate G-code for printing this ribbon pattern by tracing orthogonal families of curves along smooth curvature-aligned direction fields [64, 65].

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 almost always positive in the direction of the ribbons. We circumvent this limitation by introducing a custom, reversible frame structure that allows us to print plastic on both sides of the stretched fabric. This simple hardware further expands the range of surfaces we can reproduce (Figure 5.11).

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.

This chapter is mainly based on the following publication:

- David Jourdan, Mélina Skouras, Etienne Vouga, Adrien Bousseau, 2022. *Computational Design of Self-Actuated Surfaces by Printing Plastic Ribbons on Stretched Fabric*. Computer Graphics Forum (Proc. EUROGRAPHICS, conditionally accepted).

5.2 Related work

Out of the various methods for designing deployable structures already introduced in chapter 2, the inflatable structures of Panetta et al. [93], present an interesting case due to their similar approach to solving the inverse design problem. 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 5.3.1 and Figure 5.4).

Surface parameterization is a classic problem in geometry processing [55], with many available algorithms that seek to minimize angle [112, 121] or length distortion [160], 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 [53, 66], 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 [3]. Closest to our approach is Panetta et al. [93]’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 [79], a classic and easily-extendable local-global algorithm for distortion-minimizing parameterization. See Section 5.3.2 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 [82, 85], knitting paths [88], microstructures [142], weaves [106, 150] and nets [110], fiber-reinforced composites [19], and the aforementioned inflatable structures [93]. 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. [148] for a review of some of these alternatives. We chose the method of Knöppel et al. [65], 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.

5.3 Programming intrinsic and extrinsic curvature

In this section, we first describe the mechanics of plastic ribbons embedded in stretched fabric, then we derive the flattening formulation in the continuous setting, before discretizing it. We then show how to implement the parameterization algorithm and how to find the optimal thickness to print so as to match a given normal curvature.

5.3.1 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.

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

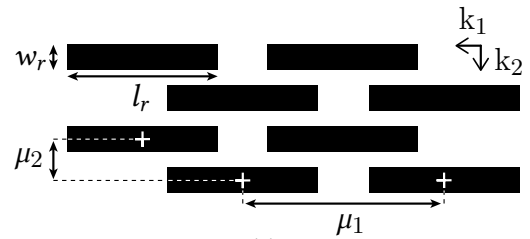


Fig. 5.3: Ribbon pattern.

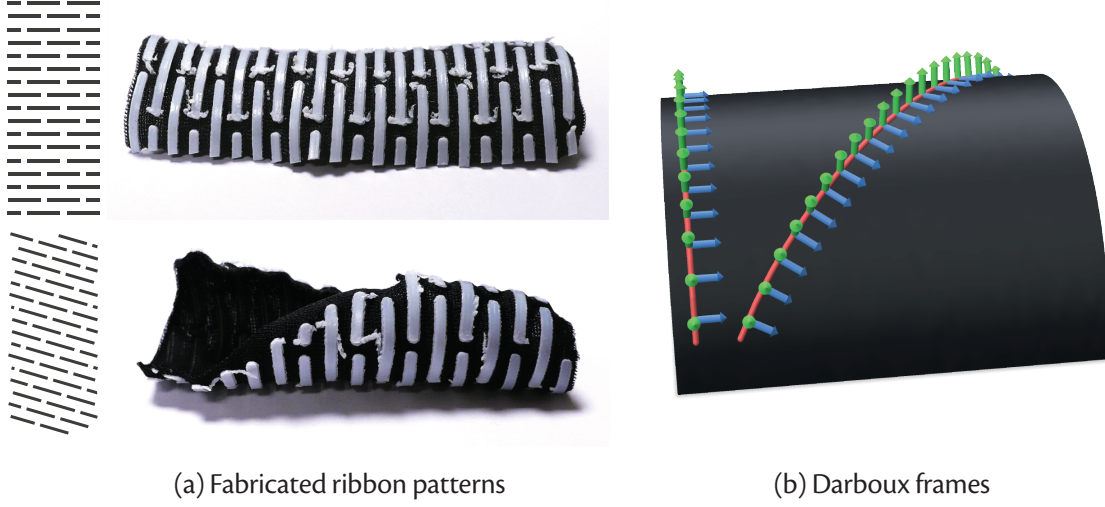


Fig. 5.4: Due to the bilayer effect, parallel plastic ribbons roll to form a cylinder (a, top). Attempting to orient the ribbons away from the direction of maximum curvature still yields a cylinder as ribbons resist torsion (a, bottom). On a curved surface, curvature lines are the only curves with zero geodesic torsion, as visualized by sliding a Darboux frame along the curve and observing its rotation around the curve tangent (b, after [56]).

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. 5.4a). Based on these two observations, we align the ribbon pattern (horizontal axis in Fig. 5.3) in the direction of maximum magnitude principal curvature k_1 , which maximally exploits the bilayer effect and minimizes torsion along the ribbons since curvature lines are characterized by vanishing geodesic torsion [16, 33, 56] (Fig. 5.4b). It

follows that the transverse direction of the pattern aligns to the other principal curvature direction 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 5.3.2).
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 experiments (Section 5.3.5).

5.3.2 Flattening with bounded directional scaling

Let the mapping $\varphi: \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$ describe our (given) target deployed surface \mathcal{S} , and let $\bar{\varphi}: \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^2$ 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 $k_1(u, v), k_2(u, v)$ for the vector fields $\Omega \rightarrow \mathbb{R}^3$ in the direction of maximum-magnitude and minimum-magnitude principal curvature of \mathcal{S} .

We seek a flattening $\bar{\varphi} \circ \varphi^{-1}: \mathcal{S} \rightarrow \bar{\mathcal{S}}$ with the following properties: (1) the principal stretch directions are aligned with k_1, 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 5.3.4, 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 $J_{\bar{\varphi}} = d\bar{\varphi} \in \mathbb{R}^{2 \times 2}$ 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 $d(\bar{\varphi} \circ \varphi^{-1})$ has the

form

$$d(\bar{\varphi} \circ \varphi^{-1}) = R_{\bar{S}} \Sigma R_S^T, \forall (u, v) \in \Omega, \quad (5.1)$$

where $R_{\bar{S}}(u, v)$ is an arbitrary rotation matrix; $R_S = \begin{bmatrix} \hat{k}_1 & \hat{k}_2 \end{bmatrix}_{3 \times 2}$ rotates the Euclidean plane to the tangent plane of 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 (5.1) 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

$$\min_{\bar{\varphi}, \sigma_1, \sigma_2, R_{\bar{S}}} \int_{\Omega} \|J_{\bar{\varphi}} - R_{\bar{S}} \Sigma R_S^T J_{\varphi}\|_F^2 \, dudv, \quad (5.2)$$

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

While this formulation is reminiscent of other flattening methods with bounded scaling [3, 93], unique to our approach is the use of the fixed matrix R_S to strictly constrain the directions of maximal strain. In contrast, Panetta et al. [93] 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. 5.4) demands close curvature alignment and limits the usefulness of this tradeoff. Moreover, hard constraints allow the use of an ARAP-style local-global solver (described next), which is more efficient (parallelizes trivially) and easier to implement than the black-box nonlinear optimization using Newton's method that would be required for soft constraints.

5.3.3 Discrete formulation

We discretize (5.2) using triangular elements. To this end, we represent the surfaces S and \bar{S} using triangle meshes having same number of vertices n_V and faces n_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.

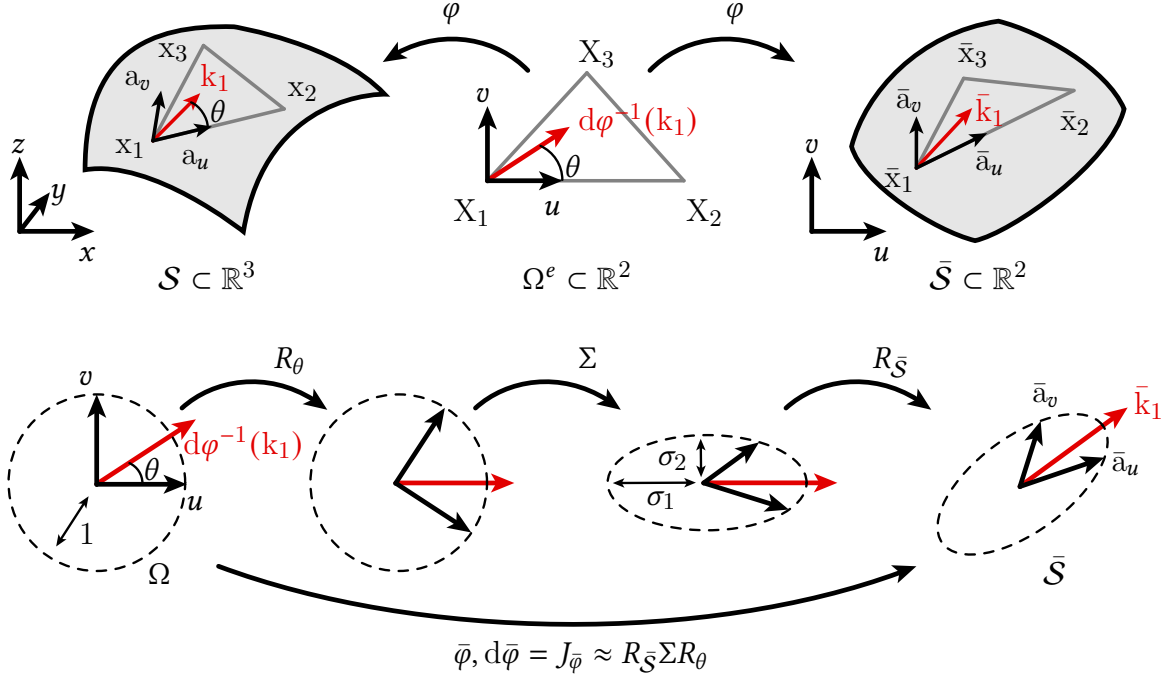


Fig. 5.5: Overview of the flattening algorithm. Each triangle of the target mesh S is first mapped to the 2D plane by aligning its (x_1, x_2) edge with the u -axis of our 2D Cartesian coordinate system. We then compute its mapping $\bar{\varphi}$ to the flattened configuration \bar{S} using a local/global algorithm (top). This mapping $\bar{\varphi}$ is built so as to maximally stretch the surface S along its direction of maximum curvature k_1 (bottom).

Note that in practice we do not need to build a full (and consistent) parametrization of the surface 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 S with 3D vertex coordinates $x_{1 \leq k \leq 3}^e$ as a 2D triangle of same shape and size with vertex coordinates X_k^e , whose edge vector $X_2^e - X_1^e$ is aligned with the u -axis of our 2D Cartesian coordinate system (see Figure 5.5). We can then express the Jacobian J^e as $J^e = [\bar{x}_2^e - \bar{x}_1^e \quad \bar{x}_3^e - \bar{x}_1^e][X_2^e - X_1^e \quad X_3^e - X_1^e]^{-1}$. Moreover, on each triangle, the parameterization of S is now an isometry, with $R_{\bar{S}}^T J_\varphi = R_\theta$ a 2×2 rotation matrix by angle θ , where θ is the angle from $d\phi^{-1}k_1$ to the u -axis.

We can now rewrite our optimization problem (5.2) as

$$\min_{\bar{x}, \sigma_1, \sigma_2, R} \underbrace{\sum_{i=1}^{n_T} \|J^i(\bar{x}) - R_{\bar{S}}^i \Sigma^i R_\theta^i\|_F^2 A_i}_{E(\bar{x}, \sigma_1, \sigma_2, R)} \quad (5.3)$$

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

To solve problem (5.3), we adapt the local/global optimization algorithm proposed by Liu et al. [79] and alternate between local steps in which we optimize the per-triangle rotations and stretches $R_{\bar{S}}^e$ and Σ^e , and global steps where we minimize the cost function E over the positions \bar{x} while keeping all the matrices $R_{\bar{S}}^e$ 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

$$(\tilde{\sigma}_1^e, \tilde{\sigma}_2^e) = \underset{\substack{(\sigma_1, \sigma_2) \\ \sigma_1^{\min} \leq \sigma_1 \leq \sigma_1^{\max}, \\ \sigma_2^{\min} \leq \sigma_2 \leq \sigma_2^{\max}}}{\operatorname{argmin}} \underbrace{\|J^e(\bar{x}) - R_{\bar{S}}^e \Sigma^e R_{\theta}^e\|_F^2}_{E(\sigma_1, \sigma_2)}. \quad (5.4)$$

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 (5.4) as

$$\begin{aligned} E(\sigma_1, \sigma_2) &= \|(R_{\bar{S}}^e)^T (J^e - R_{\bar{S}}^e \Sigma^e R_{\theta}^e) (R_{\theta}^e)^T\|_F^2 \\ &= \|(R_{\bar{S}}^e)^T J^e (R_{\theta}^e)^T - \Sigma^e\|_F^2 \\ &= \|R_{\theta}^e (J^e)^T R_{\bar{S}}^e - \Sigma^e\|_F^2. \end{aligned} \quad (5.5)$$

$E(\sigma_1, \sigma_2)$ is thus equal to the sum of the squared entries of $R_{\theta}^e (J^e)^T R_{\bar{S}}^e - \Sigma^e$:

$$E(\sigma_1, \sigma_2) = (\sigma_1 - [R_{\theta}^e (J^e)^T R_{\bar{S}}^e]_{11})^2 + (\sigma_2 - [R_{\theta}^e (J^e)^T R_{\bar{S}}^e]_{22})^2 + C \quad (5.6)$$

where C is constant with respect to σ_1 and σ_2 . The minima subject to $\sigma_i^{\min} \leq \sigma_i^e \leq \sigma_i^{\max}$ are then given by

$$\sigma_i^e = \begin{cases} \sigma_i^{\min} & \text{if } \hat{\sigma}_i^e < \sigma_i^{\min}, \\ \hat{\sigma}_i^e & \text{if } \sigma_i^{\min} \leq \hat{\sigma}_i^e \leq \sigma_i^{\max}, \\ \sigma_i^{\max} & \text{if } \sigma_i^{\max} < \hat{\sigma}_i^e, \end{cases} \quad i = \{1, 2\}, \quad (5.7)$$

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

Computation of rotation matrices $R_{\bar{S}}^e$. Minimizing E with respect to $R_{\bar{S}}^e$ amounts to solving

$$\tilde{R}_{\bar{S}}^e = \operatorname{argmin}_{R_{\bar{S}}^e \in SO(2)} \left\| J^e - R_{\bar{S}}^e \Sigma^e R_{\theta}^e \right\|_F^2. \quad (5.8)$$

which is known as the Orthogonal Procrustes problem [113], the following proof is given here for completeness but can be found in a similar form in [44, Chapter 4]:

Expanding the right hand side of equation (5.8) gives us

$$\begin{aligned} \left\| J^e - R_{\bar{S}}^e \Sigma^e R_{\theta}^e \right\|_F^2 &= \operatorname{tr} \left((J^e - R_{\bar{S}}^e \Sigma^e R_{\theta}^e)^t (J^e - R_{\bar{S}}^e \Sigma^e R_{\theta}^e) \right) \\ &= \operatorname{tr} \left((J^e)^t J^e \right) - 2 \operatorname{tr} \left((J^e)^t R_{\bar{S}}^e \Sigma^e R_{\theta}^e \right) \\ &\quad + \operatorname{tr} \left((R_{\bar{S}}^e \Sigma^e R_{\theta}^e)^t R_{\bar{S}}^e \Sigma^e R_{\theta}^e \right). \end{aligned} \quad (5.9)$$

Discarding terms that do not depend on $R_{\bar{S}}^e$ we can write

$$\begin{aligned} \tilde{R}_{\bar{S}}^e &= \operatorname{argmax}_{R_{\bar{S}}^e \in SO(2)} \operatorname{tr} \left((J^e)^t R_{\bar{S}}^e \Sigma^e R_{\theta}^e \right) \\ &= \operatorname{argmax}_{R_{\bar{S}}^e \in SO(2)} \operatorname{tr} \left(R_{\bar{S}}^e \Sigma^e R_{\theta}^e (J^e)^t \right) \end{aligned} \quad (5.10)$$

We use the signed version of the singular value decomposition of $\left(\Sigma^e R_{\theta}^e (J^e)^t \right)^t = USV^t$ such that $|S_1| < |S_2|$ with S_1 being possibly negative and $UV^t \in SO(2)$ [79]

$$\begin{aligned} \operatorname{tr} \left(R_{\bar{S}}^e \Sigma^e R_{\theta}^e (J^e)^t \right) &= \operatorname{tr} \left(R_{\bar{S}}^e V S U^t \right) \\ &= \operatorname{tr} \left(U^t R_{\bar{S}}^e V S \right) \\ &= S_1 H_{11} + S_2 H_{22}, \quad H = U^t R_{\bar{S}}^e V \end{aligned} \quad (5.11)$$

$H \in SO(2)$ therefore $H_{11} = H_{22} = \cos \theta$ for some θ in \mathbb{R}

The maximum of $\operatorname{tr} \left(R_{\bar{S}}^e \Sigma^e R_{\theta}^e (J^e)^t \right)$ is thus reached when $H = I$, i.e.

$$\tilde{R}_{\bar{S}}^e = UV^t, \quad USV^t = J^e (R_{\theta}^e)^t \Sigma^e \quad (5.12)$$

Computation of 2D node positions \bar{x} . Following Liu et al. [79] and denoting $M(e) = R_{\bar{S}}^e \Sigma^e R_{\theta}^e$, we rewrite our energy E using cotangent weights as

$$E = \sum_{(i,j) \in \mathcal{H}e} \frac{\cot \alpha_{ij}}{2} \left\| \bar{x}_i - \bar{x}_j - M(\mathcal{T}_{(i,j)})(X_i - X_j) \right\|^2, \quad (5.13)$$

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 (5.13) with respect to \bar{x} can then be obtained by solving the linear system

$$\begin{aligned} & \sum_{j \in \mathcal{N}(i)} [\cot \alpha_{ij} + \cot \alpha_{ji}] (\bar{x}_i - \bar{x}_j) \\ &= \sum_{j \in \mathcal{N}(i)} [\cot \alpha_{ij} M(\mathcal{T}_{(i,j)}) + \cot \alpha_{ji} M(\mathcal{T}_{(j,i)})] (X_i - X_j) \end{aligned} \quad (5.14)$$

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

5.3.4 Implementation details

Computation of curvature directions k_1 . We compute the directions of maximum curvature k_1 on the target mesh \mathcal{S} using the method of Knöppel et al. [64] 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 5.1).

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 limit spacing value, we deduce $\sigma_2^{\max} \approx 1.51$ using Equation 5.15 described in Section 5.4.1. For the lower bound, we set the value $\sigma_2^{\min} = 1.3$ as a safeguard against fusion of neighboring ribbons. Along 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 $\bar{x}, \sigma_1, \sigma_2$ and R . We initialize the vertex positions \bar{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* [105] (as implemented within libigl [59]). The initial stretch values σ_1^e, σ_2^e are set to 1. The rotations matrices $R_{\mathcal{S}}^e$ are initialized using Formula (5.12).

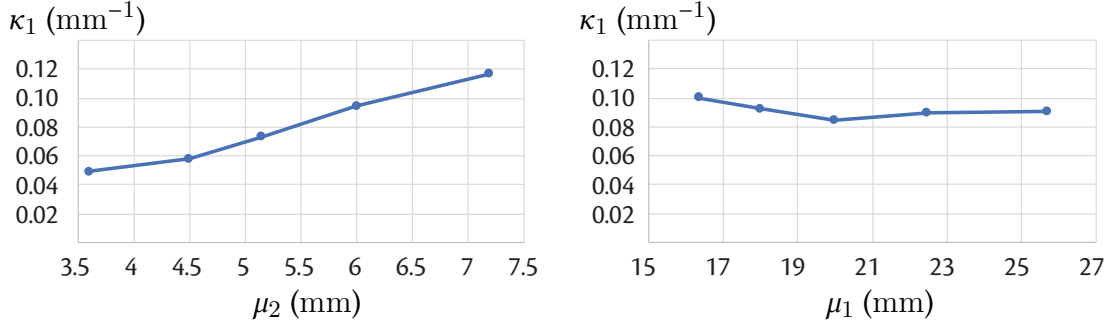


Fig. 5.6: The curvature κ_1 of our metamaterial varies not only as a function of plastic thickness τ , but also as a function of spacing μ_2 in the transverse direction (left, with $\mu_1 = 22.5\text{mm}$ and $\tau = 0.5\text{mm}$). In contrast, varying spacing μ_1 in the ribbon direction has little impact (right, with $\mu_2 = 6\text{mm}$).

Update scheme and termination. We alternate between local computations of R , σ_1 and σ_2 and global computation of \bar{x} (in this order). The loop updating R , σ_1 and σ_2 is run twice before switching to \bar{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 5.3 is nonzero. See Table 5.1 for detailed statistics about the behavior of the optimization in practice.

5.3.5 Controlling normal curvature

As discussed in the previous section, we align the ribbon pattern along the direction of maximum principal curvature 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. In chapter 3, we 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.

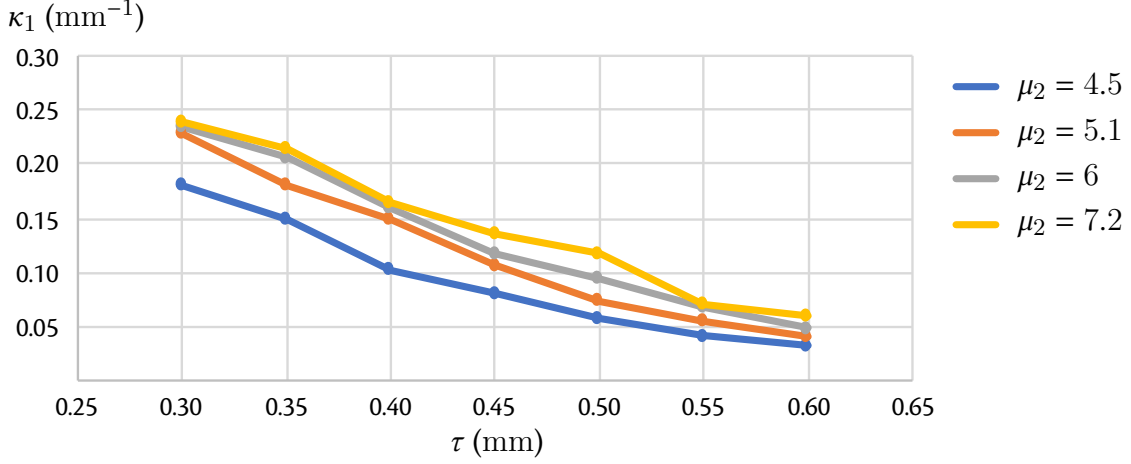


Fig. 5.7: Curvature κ_1 of our metamaterial as a function of plastic thickness τ and spacing μ_2 (in mm).

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 5.6. 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 κ_1 direction of the resulting cylinders, as reported in Figure 5.7. 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. While we only measured this data on cylindrical shapes produced by parallel patterns, our results demonstrate its generalization to more complex shapes (Figure 5.1, Figure 5.11). For many surfaces, the smooth curvature fields extracted by the methods of Knöppel et al. [64] are locally parallel.

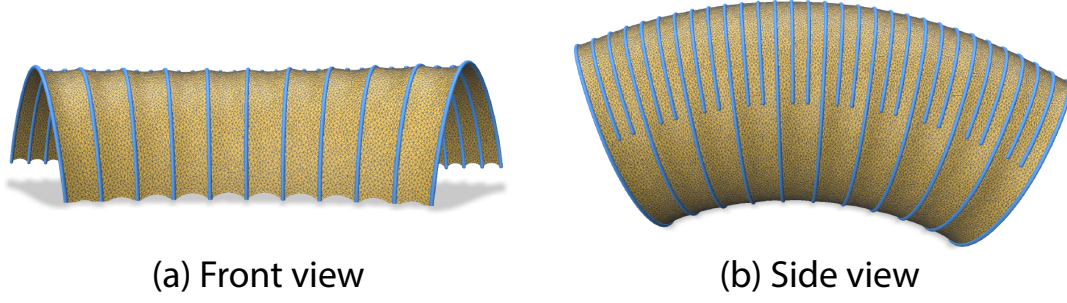


Fig. 5.8: The rod-based model from chapter 3 does not account for the width of the plastic ribbons. When attempting to simulate a similar pattern as in Fig. 5.2, the surface bends properly along the ribbons due to the bilayer effect (a), but it contracts uniformly in the direction transverse to the ribbons, 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.

We also considered physical simulation to optimize the ribbon parameters such that they reproduce the target shape. However, our experiments with the rod-based model presented in chapter 3 revealed that it 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. 5.15 below). Rod-shell coupling models that assume that the ribbons are infinitesimally wide fail to capture this phenomenon. Fig. 5.8 shows the result of our rod-based model when simulating a similar pattern as in Fig. 5.2. 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. The more complex model presented in chapter 4 might be able capture these important effects, however we have not yet fully tested it against a variety of printed samples and therefore cannot guarantee that it is capable of computing the correct curvature values for the whole range of parameters.

Front-and-back asymmetry 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

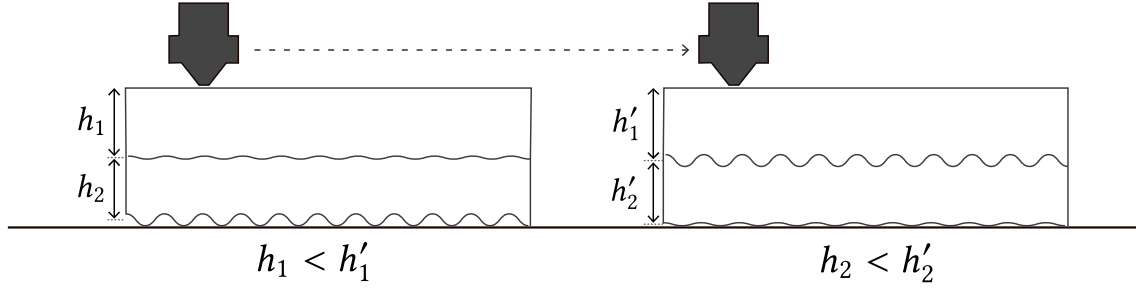


Fig. 5.9: Illustration of the front-and-back asymmetry of the textile: even though the nozzle is at the same height, the amount of extruded plastic can be bigger if the surface is rougher because the extruded plastic generally fills up all the available volume, including the small crevices in the surface of the fabric.

ribbons, when printed on the stitch front, had a 50 to 60 % higher curvature than the ones printed on the stitch back. This could be due to several factors, one explanation might be that the bending resistance of the fabric is stronger in one direction than in the other as was measured in chapter 4 (section 4.4.3), but in general the stiffness of the fabric is negligible compared to that of the plastic. Another explanation could be a geometric difference between the two sides of the fabric, where one side has a rougher surface than the other, causing the 3D printer to print a bigger thickness of fabric on the rougher side (see Fig. 5.9).

To account for this difference in curvature, we 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.

5.4 Ribbon layout

We are now equipped with a flattened surface $\bar{\mathcal{S}}$, along with the directions $\bar{k}_i = d(\bar{\varphi} \circ \varphi^{-1})k_i$ 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 5.3.1), and compute curves on $\bar{\mathcal{S}}$ that trace the centerline of each ribbon. Together with the thickness τ computed in Section 5.3.5, these parameters and curves fully determine the structure to be fabricated.

5.4.1 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.

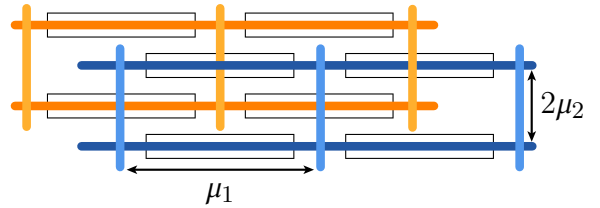
Given the fabric pre-stretch s , we compute the values of spacing μ_1 and μ_2 to achieve the target stretching values σ_1 and σ_2 by considering individual stripes of ribbon dashes. More specifically, we assume that, after we release the fabric, a periodic motif of initial length μ_1 contracts to an 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

$$\mu_1 = l_r \frac{s-1}{s-\sigma_1} \sigma_1, \quad \mu_2 = w_r \frac{s-1}{s-\sigma_2} \sigma_2. \quad (5.15)$$

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{k}_1 might prevent Equation (5.15) from holding for small values of μ_2 . Nevertheless, we found this effect to be negligible once the bound σ_2^{\min} on σ_2 was enforced during optimization.

5.4.2 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{S} so that their local spacing agrees with the target stretching values σ_1 and σ_2 according to Equation (5.15). 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{S} using the stripe pattern algorithm of Knöppel et al. [65], 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{k}_1 and \bar{k}_2 , respectively, to form one grid of our pattern. The output of the algorithm of Knöppel et al. [65] is two S^1 -valued unit complex fields over \bar{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{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.

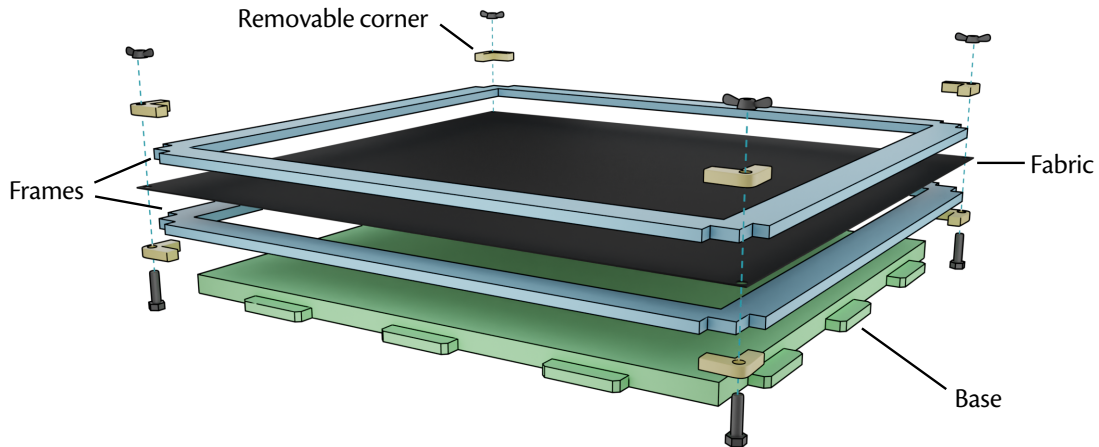


Fig. 5.10: Our frame structure is composed of two rigid frames, in between which the fabric is clamped in tension. These frames sit on a base that attaches to the printer bed. The four corners of the frames can be removed to be fixed to the sheet of fabric before stretching.

5.5 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 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 5.10 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.

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 spandex material (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.

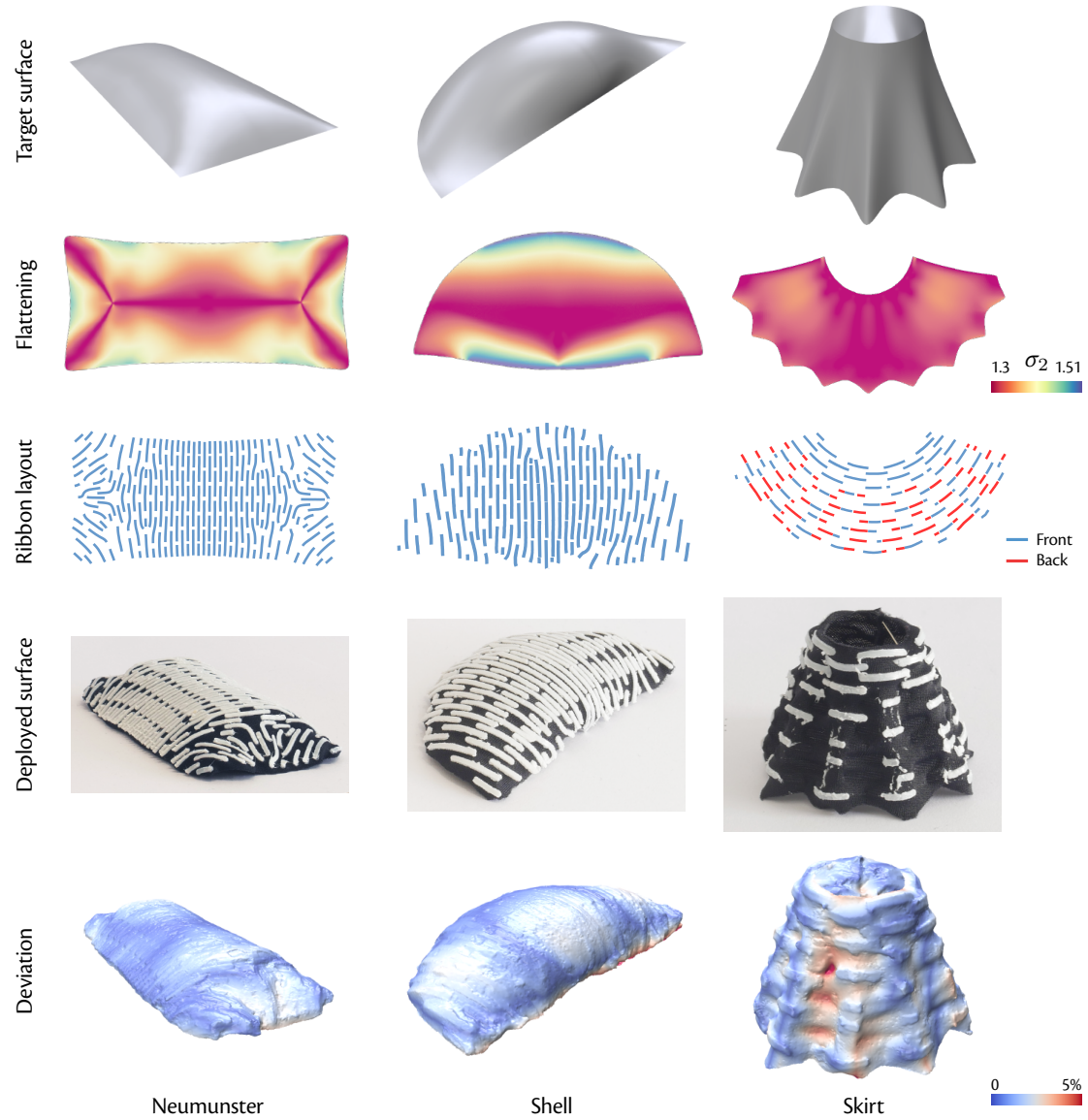


Fig. 5.11: Representative results produced with our method, including freeform doubly-curved surfaces (*Neumunster* and *Shell*) and a nearly-developable surface achieved by printing on both sides of the fabric (*Skirt*). We visualize the deviation of the scanned fabricated shape from the target surface, expressed as percentages of the bounding box diagonal of the target.

5.6 Results and discussion

Figure 5.1, Figure 5.11 and Figure 5.14 illustrate results produced with our method, including several architectural models and fashion items. Note that all these shapes deployed without manual placement of their boundaries, except the ones shown in Figure 5.13 and Figure 5.14 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 5.1) and nearly developable surfaces (*Skirt* in Figure 5.11). 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. 5.12 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 (k_2 close to 0), metric frustration also provides a way to reproduce shapes with moderate positive and negative Gaussian curvature.

We also visualize for each result in Figure 5.11 its deviation from the target surface: we reconstruct the geometry of the deployed structure from a photogrammetry scan of the fabricated object [22], 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 5.1 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

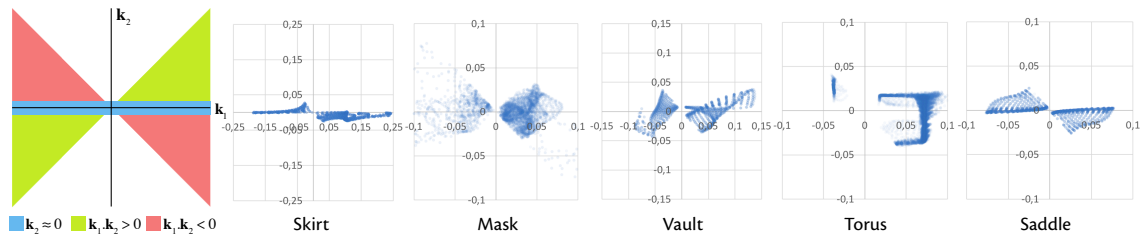


Fig. 5.12: Visualization of the two principal curvature values sampled over some of our target surfaces. While we expect our method to be most effective at controlling extrinsic curvature where the target shape is nearly-developable ($k_2 \approx 0$, such as throughout the *Skirt*), most of the surfaces we reproduced also exhibit positive and negative Gaussian curvature (*Mask*, *Vault* and *Torus*), or even solely negative Gaussian curvature (*Octopus Saddle*).

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

Table 5.1: For each result shown in this chapter we report the computation time, curvature field smoothing λ , average and maximum deviation between the scanned fabricated shape and the target (99th percentile to remove scanning outliers), average deviation of the ribbon direction from the target direction 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. 5.1) and *Vault* (Fig. 5.14). We note that these two models contain an umbilical region in their center, where the direction field exhibits a singularity.

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 [93, 95]. Table 5.1 also provides the average deviation from the prescribed direction 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 bounded by the fabric pre-stretch factor (which is $s = 1.6$ for the knitted fabric 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 [53, 114].) Although our flattening algorithm is



Fig. 5.13: 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).

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 5.3): 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. 5.1) and *Vault* (Fig. 5.14), which feature prominent umbilic regions, also have the largest shape deviation among our examples (Table 5.1). A potential future improvement would be to relax the curvature-alignment constraint (similar to the soft penalty approach of Panetta et al. [93]), 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 [38].)

Finally, our method solves the inverse design task purely geometrically, without simulat-



Fig. 5.14: Architectural model (*Vault*) and fashion items (*Skirt* and *Mask*) prototyped with our method.

ing the complex interactions between the elastic fabric and the plastic ribbons, nor the effect of external forces like gravity or additional load. Figure 5.13 and Figure 5.14 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 in a matter of seconds when physics-based optimization often takes hours. This allows the method to be usable as part of a design loop where users can quickly visualize the resulting pattern and make edits if necessary. 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.

CHAPTER 6

CONCLUSION

In this thesis, I presented and used a method consisting in extruding plastic curves onto pre-stretched fabric so as to create *self-shaping textiles* which automatically deploy when the underlying fabric is released. Two distinct metamaterials made out of repetitive and controllable patterns were presented, first a metamaterial arising from a pattern of stars which we exploited to create self-shaping architectural models, and then a metamaterial composed of parallel ribbons in which we leveraged the stretch in-between the ribbons and the bilayer effect at the ribbon-fabric interface to program the intrinsic and extrinsic curvature of the deployable surfaces.

In terms of the computational tools developed, they can be classified into two distinct design workflows. First, chapters 3 and 4 proposed a direct form-finding workflow in which the user specifies a pattern to be printed on stretched fabric and can then pre-visualizes the deployed shape which rises from the specified pattern before actually printing it. A direct workflow can be interesting in early stages of the design process when the user is exploring possible shape ideas, we apply it to the architectural modelmaking context.

Then, in Chapter 5, an inverse design tool was presented. The workflow was different: the user was first asked to specify a target shape to be reproduced, and the system found an optimal layout to be printed so as to obtain a deployed shape similar to the input. The inverse design workflow might seem more productive because the designer does not need to go through a trial-and-error process to obtain a specific shape, but in reality some of their requirements might not be satisfied by the inverse solver. The proposed inverse design tool was fast enough so that users would have no issue iterating on the input shape if necessary.

Both types of computational design workflows have been presented and studied, because I believe both have their merits and use cases. In a way, they only fundamentally differ

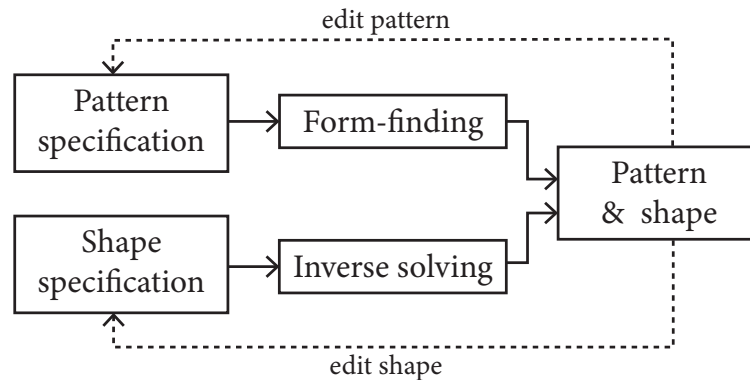


Fig. 6.1: Integrating the direct and inverse workflows: the user can start from either the shape or the pattern, and make edits on either of these two.

from their starting point in the design process: in the direct workflow, a user starts from a pattern to be printed and then iterates on that pattern until a satisfying result is found; in the inverse workflow they start from the shape to be reproduced and then modify it if the resulting pattern from the optimization is not satisfying. An interesting development would be to be able to *join* both the direct and the inverse design workflows in a system where the user can start from either the pattern or the target shape and then have the possibility to edit either one or the other and see these edits propagate to the other end (see Fig. 6.1). Such a workflow would suit both artists who wish to obtain a certain aesthetic result with printed curves, and designers who focus more on the specific shape they want to create, and would allow both types of users to make edits on either end.

Exciting potential directions for improving the simulation of printed-on-fabric deployable structures include the ongoing work presented in Chapter 4 which could be expanded into a tool that would allow users to simulate the deployment of arbitrary patterns printed on stretched fabric instead of just the tiling of stars on which we focused in Chapter 3. The goal of Chapter 4 was to circumvent some of the limitations of the form-finding method of Chapter 3 by explicitly modeling the impact of the width of printed curve, as well as modeling the orthotropic and nonlinear response of the pre-stretched textile.

Modeling the imprecisions in the printing process would be also greatly beneficial as a variety of defects can occur, either because the plastic sometimes does not adhere well to fabric (sometimes due to under-extrusions), or because it leaks from one printed element to the next as the printing head moves over the surface (over-extrusions,



as in inset). These imprecisions could be modeled in an explicit way by using a thermo-mechanical simulation of the plastic extrusion process to more precisely understand the interaction between the extruded plastic and the textile surface, or in an implicit way by using statistical tools to assess the impact of unknown factors such as, e.g. the variation in the dimensions of the printed curves, and by propagating such uncertainty within the simulation of the fabric-plastic assembly.

Many of the imprecisions we had to deal with were also amplified because we were working at quite a small scale. The commercial 3D printer we used restricted us to 20 cm-wide designs at best, which means we had to print very fine details in order for the metamaterials to assume detailed curvature variations. Being able to create larger scale self-shaping textiles would therefore be an interesting avenue for future work. It would be possible, for example, to mount an extruder on a larger 2D CNC machine so as to increase the available footprint, or to apply the same principles developed in this thesis but on slightly different fabrication techniques such as gluing plywood panels to a large pre-stretched latex sheet [12] or embedding flexible rods inside a knitted textile [7].

An interesting challenge which would arise when fabricating larger shapes is how to make a soft material such as fabric withstand its own weight. Traditionally, textiles in architecture are often kept pre-stretched to prevent them from sagging, but in our case it is the release of the membrane tension which gives the textile its shape. A possible solution could be to stiffen the fabric by coating it with a material such as concrete, a technique that has been successfully employed by Popescu et al. [102]. Another solution could be to automatically generate support structures, a problem that was the topic of my master's thesis, and for which I developed an algorithm based on a sparsity-inducing norm to find a small set of poles that would support a tensile structure [62].

BIBLIOGRAPHY

- [1] Asterios Agkathidis, Yorgos Berdos, and André Brown, 2019. “Active membranes: 3D printing of elastic fibre patterns on pre-stretched textiles”. *International Journal of Architectural Computing*, vol. 17, no. 1, pages 74–87. doi:10.1177/1478077118800890. Cited pages 22 and 27.
- [2] Asterios Agkathidis and Güzden Varinlioğlu, Jan. 2022. “4D printing on textiles: Developing a file to fabrication framework for self-forming, composite wearables”. *Advances in Product Design Engineering. Management and Industrial Engineering*. Cited pages 3, 24, and 27.
- [3] 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*, vol. 115, no. 28, pages 7206–7211. doi:10.1073/pnas.1804702115. Cited pages 2, 18, 21, 66, 68, and 72.
- [4] Sean Ahlquist, Ali Askarinejad, Rizkallah Chaaraoui, Ammar Kalo, Xiang Liu, and Kavan Shah, 2014. “Post-forming composite morphologies: Materialization and design methods for inducing form through textile material behavior”. In *Proceedings of the 34th Annual Conference of the Association for Computer Aided Design in Architecture*, ACADIA '14, pages 268–276. Cited page 22.
- [5] Sean Ahlquist and Achim Menges, 2013. “Frameworks for computational design of textile micro-architectures and material behavior in forming complex force-active structures”. In *Proceedings of the 33rd Annual Conference of the Association for Computer Aided Design in Architecture*, ACADIA '13, pages 281–292. Cited page 12.
- [6] Sally Al-Badry, Yorgos Berdos, Katya Bryskina, and Cesar Cheng. “Lobster shell” [online], 2016. URL: https://issuu.com/katyabryskina/docs/the_lobster [Accessed 2022-01-09]. Cited pages 23 and 24.
- [7] Lotte Aldinger, Georgia Margariti, Axel Körner, Seiichi Suzuki, and Jan Knippers, Jul. 2018. “Tailoring self-formation: Fabrication and simulation of membrane-actuated stiffness gradient composites”. In *Proceedings of the IASS Symposium 2018*. Cited pages 22 and 93.
- [8] 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, pages 1–12. doi:10.1145/3173574.3173834. Cited pages 20 and 21.

[9] Changyeob Baek, Andrew O. Sageman-Furnas, Mohammad K. Jawed, and Pedro M. Reis, 2018. “Form finding in elastic gridshells”. *Proceedings of the National Academy of Sciences*, vol. 115, no. 1, pages 75–80. doi:10.1073/pnas.1713841115. Cited page 11.

[10] David Baraff and Andrew Witkin, 1998. “Large steps in cloth simulation”. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 43–54. doi:10.1145/280814.280821. Cited page 50.

[11] Olivier Baverel, Jean-François Caron, Frédéric Tayeb, and Lionel Du Peloux, 2012. “Gridshells in composite materials: Construction of a 300 m² forum for the Solidays' festival in Paris”. *Structural Engineering International*, vol. 22, no. 3, pages 408–414. hal-01157370. Cited page 11.

[12] Yorgos Berdos, Asterios Agkathidis, and Andre Brown, 2020. “Architectural hybrid material composites: Computationally enabled techniques to control form generation”. *Architectural Science Review*, vol. 63, no. 2, pages 154–164. doi:10.1080/00038628.2019.1666357. Cited pages 22 and 93.

[13] Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun, Jul. 2010. “Discrete viscous threads”. *ACM Transactions on Graphics*, vol. 29, no. 4, pages 116:1–116:10. doi:10.1145/1778765.1778853. Cited page 33.

[14] Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun, Aug. 2008. “Discrete elastic rods”. *ACM Transactions on Graphics*, vol. 27, no. 3, pages 1–12. doi:10.1145/1360612.1360662. Cited pages 33 and 50.

[15] Joseph Bertrand, Charles-François Diguët, and Joseph Puiseux, 1848. “Démonstration d'un théorème de Gauss *and* Note à l'occasion de l'article précédent *and* Sur le même théorème”. *Journal de Mathématiques Pures et Appliquées*, vol. 13, pages 80–90. URL: <https://gallica.bnf.fr/ark:/12148/bpt6k16392x>. Cited page 17.

[16] Luc Biard, Rida T. Farouki, and Nicolas Szafran, 2010. “Construction of rational surface patches bounded by lines of curvature”. *Computer Aided Geometric Design*, vol. 27, pages 359–371. doi:10.1016/j.cagd.2010.03.002. Cited page 70.

[17] Alexandre Binninger, Floor Verhoeven, Philipp Herholz, and Olga Sorkine-Hornung, 2021. “Developable approximation via Gauss image thinning”. *Computer Graphics Forum*, vol. 40, no. 5, pages 289–300. doi:10.1111/cgf.14374. Cited page 15.

[18] Mahdi Bodaghi, Reza Noroozi, Ali Zolfagharian, Mohamad Fotouhi, and Saeed Norouzi,

2019. “4D printing self-morphing structures”. *Materials (Basel)*, vol. 12, no. 8, page 1353. doi: [10.3390/ma12081353](https://doi.org/10.3390/ma12081353). Cited pages 19, 20, and 21.
- [19] Narasimha Boddeti, Yunlong Tang, Kurt Maute, David W Rosen, and Martin L Dunn, Oct. 2020. “Optimal design and manufacture of variable stiffness laminated continuous fiber reinforced composites”. *Scientific reports*, vol. 10, no. 1, pages 1–15. doi: [10.1038/s41598-020-73333-4](https://doi.org/10.1038/s41598-020-73333-4). Cited page 69.
- [20] J. William Boley, Wim M. van Rees, Charles Lissandrello, Mark N. Horenstein, Ryan L. Truby, Arda Kotikian, Jennifer A. Lewis, and L. Mahadevan, 2019. “Shape-shifting structured lattices via multimaterial 4D printing”. *Proceedings of the National Academy of Sciences*, vol. 116, no. 42, pages 20,856–20,862. doi: [10.1073/pnas.1908806116](https://doi.org/10.1073/pnas.1908806116). Cited page 21.
- [21] Sebastien J.P. Callens, Nazlı Tümer, and Amir A. Zadpoor, 2019. “Hyperbolic origami-inspired folding of triply periodic minimal surface structures”. *Applied Materials Today*, vol. 15, pages 453–461. doi: [10.1016/j.apmt.2019.03.007](https://doi.org/10.1016/j.apmt.2019.03.007). Cited pages 22 and 23.
- [22] CapturingReality. “Realitycapture” [online], 2021. URL: <https://www.capturingreality.com/> [Accessed 2022-01-09]. Cited page 86.
- [23] Hsiao-Yu Chen, Arnav Sastry, Wim M. van Rees, and Etienne Vouga, Jul. 2018. “Physical simulation of environmentally induced thin shell deformation”. *ACM Transactions on Graphics*, vol. 37, no. 4, pages 146:1–146:13. doi: [10.1145/3197517.3201395](https://doi.org/10.1145/3197517.3201395). Cited pages 49 and 52.
- [24] Tian Chen, Julian Panetta, Max Schnaubelt, and Mark Pauly, Jul. 2021. “Bistable auxetic surface structures”. *ACM Transactions on Graphics*, vol. 40, no. 4, pages 39:1–39:9. doi: [10.1145/3450626.3459940](https://doi.org/10.1145/3450626.3459940). Cited pages 13, 16, and 65.
- [25] 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*, vol. 16, no. 5. Cited page 21.
- [26] Tingyu Cheng, Guanyun Wang, Youngwook Do, Humphrey Yang, Ye Tao, Jianzhe Gu, Byoungkwon An, and Lining Yao, 2018. “Printed paper actuator: A low-cost reversible actuation and sensing method for shape changing interfaces”. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI ’18, pages 1–12. doi: [10.1145/3173574.3174143](https://doi.org/10.1145/3173574.3174143). Cited page 21.
- [27] Simon K. Chiu, Michael Paglia (Writer), and Joshua V. Hassel (Director), 2015. “Frei otto: Spanning the future”. URL: <http://www.freiottofilm.com/>. Cited page 11.

- [28] Jorge Christie, Jun. 2017. “Spatial textile hybrids: Computing a self-forming behavior”. In *Proceedings of the 3rd International Conference of Biodigital Architecture and Genetics*, pages 2–11. Barcelona. Cited pages 23 and 25.
- [29] Gabriel Cirio, Jorge Lopez-Moreno, David Miraut, and Miguel A. Otaduy, Nov. 2014. “Yarn-level simulation of woven cloth”. *ACM Transactions on Graphics*, vol. 33, no. 6, pages 207:1–207:11. doi:10.1145/2661229.2661279. Cited page 50.
- [30] Dorothee Clasen, Oliver Köneke, Sascha Praet, and Moritz Wallasch, 2016. “Sonogrid”. Purmundus Challenge 2016, special mention. Cited page 23.
- [31] Anders Holden Deleuran, Michel Schmeck, Gregory Charles Quinn, Christoph Gengnagel, Martin Tamke, and Mette Ramsgaard Thomsen, Aug. 2015. “The tower: Modelling, analysis and construction of bending active tensile membrane hybrid structures”. In *Proceedings of IASS Annual Symposium 2015*, vol. 23, pages 1–13. Cited page 12.
- [32] Erik Demaine and Tomohiro Tachi, Jul. 2017. “Origamizer: A practical algorithm for folding any polyhedron”. In *Proceedings of the 33rd International Symposium on Computational Geometry*, SoCG 2017, pages 1–15. doi:10.4230/LIPIcs.SoCG.2017.34. Cited page 15.
- [33] Manfredo P. do Carmo, 1976. *Differential Geometry of Curves and Surfaces, 2nd edition*. Prentice-Hall, Englewood Cliffs, NJ. Cited pages 5, 11, and 70.
- [34] Lionel du Peloux, Frédéric Tayeb, Olivier Baverel, and Jean-François Caron, Sep. 2013. “Faith can also move composite gridshells”. In *Proceedings of IASS Annual Symposium 2013*, pages 1–7. Wrocław. Cited page 11.
- [35] Simon Duenser, Roi Poranne, Bernhard Thomaszewski, and Stelian Coros, Jul. 2020. “Robo-cut: Hot-wire cutting with robot-controlled flexible rods”. *ACM Transactions on Graphics*, vol. 39, no. 4, pages 98:1–98:15. doi:10.1145/3386569.3392465. Cited page 15.
- [36] Nick Dunn, 2014. *Architectural Modelmaking*. Laurence King Publishing. Cited page 27.
- [37] Alessio Erioli and Lapo Naldoni. “informed flexible matter workshop” [online], 2017. URL: <https://www.co-de-it.com/informed-flexible-matter.html> [Accessed 2022-01-09]. Cited pages 3, 24, 27, and 45.
- [38] Gabe Fields. “Self forming structures: An exploration into 3D printing on pre-stretched fabric” [online], 2018. URL: <https://n-e-r-v-o-u-s.com/blog/?p=8011> [Accessed 2022-01-09]. Cited pages 24, 45, 47, and 88.
- [39] Akash Garg, Andrew O. Sageman-Furnas, Bailin Deng, Yonghao Yue, Eitan Grinspun, Mark Pauly, and Max Wardetzky, Jul. 2014. “Wire mesh design”. *ACM Transactions on Graphics*, vol. 33,

- no. 4, pages 66:1–66:12. doi:10.1145/2601097.2601106. Cited page 11.
- [40] Carl Friedrich Gauß, 1827. *Disquisitiones generales circa superficies curvas*. Dieterichsche Buchhandlung. URL: <https://gallica.bnf.fr/ark:/12148/bpt6k62415t>. Cited page 10.
- [41] Moritz Geilinger, Roi Poranne, Ruta Desai, Bernhard Thomaszewski, and Stelian Coros, Jul. 2018. “Skaterbots: Optimization-based design and motion synthesis for robotic creatures with legs and wheels”. *ACM Transactions on Graphics*, vol. 37, no. 4, pages 160:1–160:12. doi:10.1145/3197517.3201368. Cited page 13.
- [42] A. Sydney Gladman, Elisabetta A. Matsumoto, Ralph G. Nuzzo, L. Mahadevan, and Jennifer A. Lewis, 2016. “Biomimetic 4D printing”. *Nature Materials*, vol. 15, pages 413–419. doi:10.1038/nmat4544. Cited pages 18 and 21.
- [43] Maas Goudswaard, Abel Abraham, Bruna Goveia da Rocha, Kristina Andersen, and Rong-Hao Liang, 2020. “Fabriclick: Interweaving pushbuttons into fabrics using 3D printing and digital embroidery”. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, DIS ’20, pages 379–393. Eindhoven. doi:10.1145/3357236.3395569. Cited pages 23 and 24.
- [44] John C. Gower and Garmt B. Dijksterhuis, 2004. *Procrustes problems*. Oxford University Press. doi:10.1093/acprof:oso/9780198510581.001.0001. Cited page 75.
- [45] Itay Griniasty, Hillel Aharoni, and Efi Efrati, Sep. 2019. “Curved geometries from planar director fields: Solving the two-dimensional inverse problem”. *Physical Review Letters*, vol. 123, page 127,801. doi:10.1103/PhysRevLett.123.127801. Cited page 18.
- [46] Eitan Grinspun, Yotam Gingold, Jason Reisman, and Denis Zorin, 2006. “Computing discrete shape operators on general meshes”. *Computer Graphics Forum*, vol. 25, no. 3, pages 547–556. doi:10.1111/j.1467-8659.2006.00974.x. Cited page 52.
- [47] Eitan Grinspun, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder, 2003. “Discrete shells”. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA ’03, pages 62–67. doi:10.5555/846276.846284. Cited pages 32 and 53.
- [48] 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*, vol. 5, no. 9. Cited page 21.
- [49] 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, pages 1–10. doi:10.1145/3290605.3300267. Cited pages 18, 20,

and 21.

[50] 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, pages 1–12. doi:10.1145/3424630.3425420. Cited pages 16, 18, 20, and 21.

[51] Christophe Guberan and Carlo Clopath. “Active shoes” [online], 2016. URL: <http://www.christopheguberan.ch/active-shoes/> [Accessed 2022-01-09]. Cited pages 5, 23, 45, and 64.

[52] Ruslan Guseinov, Connor McMahan, Jesús Pérez, Chiara Daraio, and Bernd Bickel, Dec. 2020. “Programming temporal morphing of self-actuated shells”. *Nature Communications*, vol. 11, page 237. doi:10.1038/s41467-019-14015-2. Cited pages 22 and 23.

[53] Ruslan Guseinov, Eder Miguel, and Bernd Bickel, Jul. 2017. “Curveups: Shaping objects from flat plates with tension-actuated curvature”. *ACM Transactions on Graphics*, vol. 36, no. 4, pages 64:1–64:12. doi:10.1145/3072959.3073709. Cited pages 22, 23, 68, and 87.

[54] Christian Hafner and Bernd Bickel, Jul. 2021. “The design space of plane elastic curves”. *ACM Transactions on Graphics*, vol. 40, no. 4, pages 126:1–126:20. doi:10.1145/3450626.3459800. Cited pages 13 and 16.

[55] Kai Hormann, Bruno Lévy, and Alla Sheffer, 2007. “Mesh parameterization: Theory and practice”. In *ACM SIGGRAPH 2007 Courses*, SIGGRAPH ’07. doi:10.1145/1281500.1281510. Cited page 68.

[56] Emmanuel Iarussi, David Bommes, and Adrien Bousseau, May 2015. “Bendfields: Regularized curvature fields from rough concept sketches”. *ACM Transactions on Graphics*, vol. 34, no. 3, pages 24:1–24:16. doi:10.1145/2710026. Cited page 70.

[57] Alexandra Ion, Michael Rabinovich, Philipp Herholz, and Olga Sorkine-Hornung, Nov. 2020. “Shape approximation by developable wrapping”. *ACM Transactions on Graphics*, vol. 39, no. 6. doi:10.1145/3414685.3417835. Cited page 15.

[58] Sadashige Ishida, Masafumi Yamamoto, Ryoichi Ando, and Toshiya Hachisuka, Nov. 2017. “A hyperbolic geometric flow for evolving films and foams”. *ACM Transactions on Graphics*, vol. 36, no. 6, pages 199:1–199:11. doi:10.1145/3130800.3130835. Cited page 11.

[59] Alec Jacobson, Daniele Panozzo, et al., 2018. “libigl: A simple C++ geometry processing library”. URL: <https://libigl.github.io/>. Cited pages vi and 76.

[60] Caigui Jiang, Florian Rist, Helmut Pottmann, and Johannes Wallner, Nov. 2020. “Freeform quad-based kirigami”. *ACM Transactions on Graphics*, vol. 39, no. 6, pages 209:1–209:11. doi:

10.1145/3414685.3417844. Cited page 15.

[61] David Jourdan, 2021. “fabsim: Tools for fabrication & simulation”. URL: <https://github.com/DavidJourdan/fabsim> [Accessed 2022-01-09]. Cited pages 31 and 38.

[62] David Jourdan, Mélina Skouras, and Adrien Bousseau, Nov. 2018. “Optimizing support structures for tensile architecture”. In *Journées Françaises d’Informatique Graphique (JFIG)*. Cited page 93.

[63] Jungwook Kim, James A. Hanna, Myunghwan Byun, Christian D. Santangelo, and Ryan C. Hayward, 2012. “Designing responsive buckled surfaces by halftone gel lithography”. *Science*, vol. 335, no. 6073, pages 1201–1205. doi:10.1126/science.1215309. Cited pages 2, 18, 21, 24, 28, and 31.

[64] Felix Knöppel, Keenan Crane, Ulrich Pinkall, and Peter Schröder, Jul. 2013. “Globally optimal direction fields”. *ACM Transactions on Graphics*, vol. 32, no. 4, pages 59:1–59:10. doi:10.1145/2461912.2462005. Cited pages 67, 76, and 78.

[65] Felix Knöppel, Keenan Crane, Ulrich Pinkall, and Peter Schröder, Jul. 2015. “Stripe patterns on surfaces”. *ACM Transactions on Graphics*, vol. 34, pages 39:1–39:11. doi:10.1145/2767000. Cited pages 67, 69, and 82.

[66] Mina Konaković, Keenan Crane, Bailin Deng, Sofien Bouaziz, Daniel Piker, and Mark Pauly, Jul. 2016. “Beyond developable: Computational design and fabrication with auxetic materials”. *ACM Transactions on Graphics*, vol. 35, no. 4, pages 89:1–89:11. doi:10.1145/2897824.2925944. Cited pages 13 and 68.

[67] Mina Konaković-Luković, Pavle Konaković, and Mark Pauly, 2018. “Computational design of deployable auxetic shells”. In *Advances in Architectural Geometry 2018*, pages 94–111. Gothenburg. Cited pages 28 and 31.

[68] Mina Konaković-Luković, Julian Panetta, Keenan Crane, and Mark Pauly, Jul. 2018. “Rapid deployment of curved surfaces via programmable auxetics”. *ACM Transactions on Graphics*, vol. 37, no. 4, pages 106:1–106:13. doi:10.1145/3197517.3201373. Cited pages 16 and 65.

[69] Agata Kycia, 2017. “Self-shaping textiles: Potential of 3D printing for the design of performative textile structures”. URL: <https://agatakycia.com/2019/03/05/self-shaping-textiles/> [Accessed 2022-01-13]. Cited page 3.

[70] Agata Kycia, 2018. “Material form-finding of modular textile structures”. In *CA²RE: Conference for Artistic and Architectural (Doctoral) Research*, pages 331–341. Aarhus. Cited pages 24 and 27.

- [71] Agata Kycia, 2019. “Hybrid textile structures as means of material-informed design strategy”. In *CA²RE Berlin Proceedings: Conference for Artistic and Architectural (Doctoral) Research*, pages 34–35. Berlin. Cited pages 23 and 24.
- [72] Agata Kycia, Sep. 2020. “Self-shaping textiles – a material platform for digitally designed, material-informed surface elements”. In *Proceedings of the 38th eCAADe Conference*, pages 21–30. Berlin. Cited pages 23, 24, and 25.
- [73] Riccardo La Magna, Simon Schleicher, and Jan Knippers, Sep. 2016. “Bending-active plates: Form and structure”. In *Advances in Architectural Geometry 2016*. Zurich. doi:10.3218/3778-4_13. Cited page 12.
- [74] Francesco Laccone, Luigi Malomo, Jesús Pérez, Nico Pietroni, Federico Ponchio, Bernd Bickel, and Paolo Cignoni, Oct. 2019. “Flexmaps pavilion: A twisted arc made of mesostructured flat flexible panels”. In *Proceedings of the IASS Annual Symposium 2019*, pages 498–504. Barcelona. Cited page 13.
- [75] Claire Lestringant, Basile Audoly, and Dennis Kochmann, 2019. “A discrete, geometrically exact method for simulating nonlinear, elastic or non-elastic beams”. *Computer Methods in Applied Mechanics and Engineering*, vol. 361, page 112,741. doi:10.1016/j.cma.2019.112741. Cited page 33.
- [76] Claire Lestringant and Dennis M. Kochmann, May 2020. “Modeling of Flexible Beam Networks and Morphing Structures by Geometrically Exact Discrete Beams”. *Journal of Applied Mechanics*, vol. 87, no. 8. doi:10.1115/1.4046895. Cited page 33.
- [77] Wanda J Lewis, 2017. *Tension Structures: Form and behaviour, Second edition*. ICE Publishing. doi:10.1680/tsfab.61736. Cited page 11.
- [78] Yijing Li and Jernej Barbič, 2015. “Stable orthotropic materials”. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA ’14, pages 41–46. doi:10.2312/sca.20141121. Cited page 54.
- [79] 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, pages 1495–1504. Copenhagen. doi:10.5555/1731309.1731336. Cited pages 19, 66, 68, 74, and 75.
- [80] Jesse Louis-Rosenberg and Jessica Rosenkrantz. “Puzzle cell complex” [online], 2020. URL: <https://n-e-r-v-o-u-s.com/blog/?p=8521> [Accessed 2022-01-09]. Cited pages 15 and 16.
- [81] Long Ma, Ying He, Qian Sun, Yuanfeng Zhou, Caiming Zhang, and Wenping Wang, Oct.

2019. “Constructing 3D self-supporting surfaces with isotropic stress using 4D minimal hypersurfaces of revolution”. *ACM Transactions on Graphics*, vol. 38, no. 5, pages 144:1–144:13. doi:10.1145/3188735. Cited page 12.
- [82] Zhao Ma, Alexander Walzer, Christian Schumacher, Romana Rust, Fabio Gramazio, Matthias Kohler, and Moritz Bäcker, 2020. “Designing robotically-constructed metal frame structures”. *Computer Graphics Forum*, vol. 39, no. 2, pages 411–422. doi:10.1111/cgf.13940. Cited page 69.
- [83] Luigi Malomo, Jesús Pérez, Emmanuel Iarussi, Nico Pietroni, Eder Miguel, Paolo Cignoni, and Bernd Bickel, Dec. 2018. “Flexmaps: Computational design of flat flexible shells for shaping 3D objects”. *ACM Transactions on Graphics*, vol. 37, no. 6, page 241,144:241:14. doi:10.1145/3272127.3275076. Cited pages 13 and 15.
- [84] Eder Miguel, Derek Bradley, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Miguel A. Otaduy, and Steve Marschner, 2012. “Data-driven estimation of cloth simulation models”. *Computer Graphics Forum*, vol. 31, no. 2, pages 519–528. doi:10.1111/j.1467-8659.2012.03031.x. Cited page 53.
- [85] Eder Miguel, Mathias Lepoutre, and Bernd Bickel, Jul. 2016. “Computational design of stable planar-rod structures”. *ACM Transactions on Graphics*, vol. 35, no. 4, pages 86:1–86:11. doi:10.1145/2897824.2925978. Cited page 69.
- [86] Masaaki Miki, Sigrid Adriaenssens, Takeo Igarashi, and Ken’ichi Kawaguchi, 2014. “The geodesic dynamic relaxation method for problems of equilibrium with equality constraint conditions”. *International Journal for Numerical Methods in Engineering*, vol. 99, no. 9, pages 682–710. Cited page 12.
- [87] 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*, vol. 12, pages 4985–4990. doi:10.1039/C6SM00714G. Cited page 21.
- [88] Vidya Narayanan, Lea Albaugh, Jessica Hodgins, Stelian Coros, and James Mccann, Aug. 2018. “Automatic machine knitting of 3D meshes”. *ACM Transactions on Graphics*, vol. 37, no. 3. doi:10.1145/3186265. Cited page 69.
- [89] Joanna Neuß, Mirja Kreuziger, Nils Grimmelsmann, Michael Korger, and Andrea Ehrmann, Nov. 2016. “Interaction between 3D deformation of textile fabrics and imprinted lamellae”. In *Proceedings of the Aachen-Dresden-Denkendorf International Textile Conference*. Cited pages 24 and 25.

- [90] Jorge Nocedal and Stephen J. Wright, 2006. *Numerical Optimization*. Springer. Cited pages 12, 14, and 38.
- [91] Jifei Ou, Mélina Skouras, Nikolaos Vlavianos, Felix Heibeck, Chin-Yi Cheng, Jannik Peters, and Hiroshi Ishii, 2016. “Aeromorph - heat-sealing inflatable shape-change materials for interaction design”. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST ’16, pages 121–132. Tokyo. doi:10.1145/2984511.2984520. Cited page 22.
- [92] Neri Oxman and Jesse Louis-Rosenberg, 2007. “Material-based design computation: An inquiry into digital simulation of physical material properties as design generators”. *International Journal of Architectural Computing*, vol. 5, no. 1, pages 26–44. doi:10.1260/147807707780912985. Cited page 22.
- [93] Julian Panetta, Florin Isvoranu, Tian Chen, Emmanuel Siéfert, Benoît Roman, and Mark Pauly, Jul. 2021. “Computational inverse design of surface-based inflatables”. *ACM Transactions on Graphics*, vol. 40, no. 4, pages 40:1–40:14. doi:10.1145/3450626.3459789. Cited pages 13, 15, 22, 66, 68, 69, 72, 87, and 88.
- [94] Julian Panetta, Mina Konaković-Luković, Florian Isvoranu, Etienne Bouleau, and Mark Pauly, Jul. 2019. “X-shells: A new class of deployable beam structures”. *ACM Transactions on Graphics*, vol. 38, no. 4, pages 83:1–83:15. doi:10.1145/3306346.3323040. Cited page 11.
- [95] Jesús Pérez, Miguel A. Otaduy, and Bernhard Thomaszewski, Jul. 2017. “Computational design and automated fabrication of Kirchhoff-Plateau surfaces”. *ACM Transactions on Graphics*, vol. 36, no. 4, pages 62:1–62:12. doi:10.1145/3072959.3073695. Cited pages 12, 15, 24, 27, 28, 32, 39, 43, 46, 48, 64, and 87.
- [96] Jesús Pérez, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, José A. Canabal, Robert Sumner, and Miguel A. Otaduy, Jul. 2015. “Design and fabrication of flexible rod meshes”. *ACM Transactions on Graphics*, vol. 34, no. 4, pages 138:1–138:12. doi:10.1145/2766998. Cited pages 15 and 33.
- [97] Matteo Pezzulla, Norbert Stoop, Xin Jiang, and Douglas Holmes, May 2017. “Curvature-driven morphing of non-euclidean shells”. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 473, no. 2201. doi:10.1098/rspa.2017.0087. Cited page 20.
- [98] Daniel Piker, 2015. “Kangaroo physics”. URL: <http://kangaroo3d.com/> [Accessed 2022-01-09]. Cited page 12.
- [99] Stefan Pillwein, Kurt Leimer, Michael Birsak, and Przemyslaw Musialski, Jul. 2020. “On

elastic geodesic grids and their planar to spatial deployment”. *ACM Transactions on Graphics*, vol. 39, no. 4, pages 125:1–125:12. doi:10.1145/3386569.3392490. Cited page 11.

[100] Stefan Pillwein and Przemyslaw Musialski, Nov. 2021. “Generalized deployable elastic geodesic grids”. *ACM Transactions on Graphics*, vol. 40, no. 6, pages 271:1–271:15. doi:10.1145/3478513.3480516. Cited page 16.

[101] Ulrich Pinkall and Konrad Polthier, 1993. “Computing discrete minimal surfaces and their conjugates”. *Experimental mathematics*, vol. 2, no. 1, pages 15–36. Cited page 11.

[102] Mariana Popescu, Matthias Rippmann, Andrew Liew, Lex Reiter, Robert J. Flatt, Tom Van Mele, and Philippe Block, 2021. “Structural design, digital fabrication and construction of the cable-net and knitted formwork of the knitcandela concrete shell”. *Structures*, vol. 31, pages 1287–1299. doi:10.1016/j.istruc.2020.02.013. Cited page 93.

[103] Hang Jerry Qi and Mary C. Boyce, 2005. “Stress-strain behavior of thermoplastic polyurethanes”. *Mechanics of Materials*, vol. 37, no. 8, pages 817–839. doi:https://doi.org/10.1016/j.mechmat.2004.08.001. Cited page 53.

[104] Michael Rabinovich, Tim Hoffmann, and Olga Sorkine-Hornung, Jul. 2018. “Discrete geodesic nets for modeling developable surfaces”. *ACM Transactions on Graphics*, vol. 37, no. 2, pages 16:1–16:17. doi:10.1145/3180494. Cited page 10.

[105] Michael Rabinovich, Roi Poranne, Daniele Panozzo, and Olga Sorkine-Hornung, Jul. 2017. “Scalable locally injective mappings”. *ACM Transactions on Graphics*, vol. 36, no. 2. doi:10.1145/2983621. Cited page 76.

[106] Yingying Ren, Julian Panetta, Tian Chen, Florin Isvoranu, Samuel Poincloux, Christopher Brandt, Alison Martin, and Mark Pauly, Jul. 2021. “3D weaving with curved ribbons”. *ACM Transactions on Graphics*, vol. 40, no. 4, pages 127:1–127:15. doi:10.1145/3450626.3459788. Cited pages 13, 15, and 69.

[107] Ron D. Resch, 1970. “Paper and stick film”. URL: <https://vimeo.com/36122966>. Cited page 15.

[108] Victor Romero, Mickaël Ly, Abdullah Haroon Rasheed, Raphaël Charrondière, Arnaud Lazarus, Sébastien Neukirch, and Florence Bertails-Descoubes, jul 2021. “Physical validation of simulators in computer graphics: A new framework dedicated to slender elastic structures and frictional contact”. *ACM Transaction on Graphics*, vol. 40, no. 4, pages 66:1–66:19. doi:10.1145/3450626.3459931. Cited page 58.

[109] Markus Rüggeberg and Ingo Burgert, Apr. 2015. “Bio-inspired wooden actuators for large

scale applications”. *PLOS ONE*, vol. 10, no. 4, pages 1–16. [doi:10.1371/journal.pone.0120718](https://doi.org/10.1371/journal.pone.0120718). Cited pages 19 and 21.

[110] Andrew O. Sageman-Furnas, Albert Chern, Mirela Ben-Chen, and Amir Vaxman, Nov. 2019. “Chebyshev nets from commuting polyvector fields”. *ACM Transactions on Graphics*, vol. 38, no. 6, pages 172:1–172:16. [doi:10.1145/3355089.3356564](https://doi.org/10.1145/3355089.3356564). Cited pages 11 and 69.

[111] Christian D. Santangelo, 2017. “Extreme mechanics: Self-folding origami”. *Annual Review of Condensed Matter Physics*, vol. 8, no. 1, pages 165–183. [doi:10.1146/annurev-conmatphys-031016-025316](https://doi.org/10.1146/annurev-conmatphys-031016-025316). Cited page 19.

[112] Rohan Sawhney and Keenan Crane, Dec. 2017. “Boundary first flattening”. *ACM Transactions on Graphics*, vol. 37, no. 1, pages 5:1–5:14. [doi:10.1145/3132705](https://doi.org/10.1145/3132705). Cited page 68.

[113] Peter H. Schönemann, 1966. “A generalized solution of the orthogonal Procrustes problem”. *Psychometrika*, vol. 31, pages 1–10. [doi:10.1007/BF02289451](https://doi.org/10.1007/BF02289451). Cited page 75.

[114] Christian Schüller, Roi Poranne, and Olga Sorkine-Hornung, Jul. 2018. “Shape representation by zippables”. *ACM Transactions on Graphics*, vol. 37, no. 4, pages 78:1–78:13. [doi:10.1145/3197517.3201347](https://doi.org/10.1145/3197517.3201347). Cited page 87.

[115] Christian Schumacher, Steve Marschner, Markus Gross, and Bernhard Thomaszewski, jul 2018. “Mechanical characterization of structured sheet materials”. *ACM Transactions on Graphics*, vol. 37, no. 4, pages 148:1–148:15. [doi:10.1145/3197517.3201278](https://doi.org/10.1145/3197517.3201278). Cited page 47.

[116] Silvia Sellán, Noam Aigerman, and Alec Jacobson, Jul. 2020. “Developability of heightfields via rank minimization”. *ACM Transactions on Graphics*, vol. 39, no. 4, pages 109:1–109:15. [doi:10.1145/3386569.3392419](https://doi.org/10.1145/3386569.3392419). Cited page 15.

[117] Shahida Sharmin and Sean Ahlquist, 2016. “Knit architecture: Exploration of hybrid textile composites through the activation of integrated material behavior”. In *Proceedings of the 36th Annual Conference of the Association for Computer Aided Design in Architecture*, ACADIA ’16, pages 254–259. Cited page 22.

[118] Eran Sharon and Efi Efrati, 2010. “The mechanics of non-euclidean plates”. *Soft Matter*, vol. 6, pages 5693–5704. [doi:10.1039/C0SM00479K](https://doi.org/10.1039/C0SM00479K). Cited pages 17, 28, and 30.

[119] Nicholas Sharp and Keenan Crane, Jul. 2018. “Variational surface cutting”. *ACM Transactions on Graphics*, vol. 37, no. 4, pages 156:1–156:13. [doi:10.1145/3197517.3201356](https://doi.org/10.1145/3197517.3201356). Cited page 15.

[120] Nicholas Sharp et al., 2019. “Polyscope”. www.polyscope.run. Cited page vi.

[121] Alla Sheffer, Bruno Lévy, Maxim Mogilnitsky, and Alexander Bogomyakov, Apr. 2005.

- “Abf++: Fast and robust angle based flattening”. *ACM Transactions on Graphics*, vol. 24, no. 2, pages 311–330. doi:10.1145/1061347.1061354. Cited page 68.
- [122] Jonathan Richard Shewchuk, 1996. “Triangle: Engineering a 2D quality mesh generator and delaunay triangulator”. In *Selected Papers from the Workshop on Applied Computational Geometry, Towards Geometric Engineering*, pages 203–222. London. Cited page 33.
- [123] Dereje Berihun Sitotaw, Dustin Ahrendt, Yordan Kyosev, and Abera Kechi Kabish, 2020. “Additive manufacturing and textiles: State-of-the-art”. *Applied Sciences*, vol. 10, no. 15. doi:10.3390/app10155033. Cited page 23.
- [124] Emmanuel Siéfert, Etienne Reyssat, José Bico, and Benoît Roman, Jan. 2019. “Bio-inspired pneumatic shape-morphing elastomers”. *Nature Materials*, vol. 18. doi:10.1038/s41563-018-0219-x. Cited page 22.
- [125] Mélina Skouras, Bernhard Thomaszewski, Bernd Bickel, and Markus Gross, 2012. “Computational design of rubber balloons”. *Computer Graphics Forum*, vol. 31, no. 2, pages 835–844. doi:10.1111/j.1467-8659.2012.03064.x. Cited page 14.
- [126] Mélina Skouras, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, and Markus Gross, Jul. 2013. “Computational design of actuated deformable characters”. *ACM Transactions on Graphics*, vol. 32, no. 4, pages 82:1–82:10. doi:10.1145/2461912.2461979. Cited page 14.
- [127] Mélina Skouras, Bernhard Thomaszewski, Peter Kaufmann, Akash Garg, Bernd Bickel, Eitan Grinspun, and Markus Gross, Jul. 2014. “Designing inflatable structures”. *ACM Transactions on Graphics*, vol. 33, no. 4, pages 63:1–63:10. doi:10.1145/2601097.2601166. Cited page 14.
- [128] Evy Slabbinck, Seiichi Suzuki, James Solly, Anja Mader, and Jan Knippers, Sep. 2017. “Conceptual framework for analyzing and designing bending-active tensile hybrid structures”. In *Proceedings of the IASS Annual Symposium 2017*. doi:10.5281/zenodo.2625002. Cited page 12.
- [129] Enrique Nadal Soriano and Ramon Sastre Sastre, Oct. 2019. “G-shells: Flat collapsible geodesic mechanisms for gridshells”. In *Proceedings of the IASS Annual Symposium 2019*. Barcelona. Cited page 11.
- [130] Georg Sperl, Rahul Narain, and Chris Wojtan, Jul. 2020. “Homogenized yarn-level cloth”. *ACM Transactions on Graphics*, vol. 39, no. 4, pages 48:1–48:16. doi:10.1145/3386569.3392412. Cited page 50.
- [131] Michael Spivak, 1999. *A Comprehensive Introduction to Differential Geometry, Vol. 2, 3rd Edition*. Publish or Perish. Cited page 17.

- [132] Scott E. Stapleton, Dorit Kaufmann, Helga Krieger, Jan Schenk, Thomas Gries, and David Schmelzeisen, 2019. “Finite element modeling to predict the steady-state structural behavior of 4D textiles”. *Textile Research Journal*, vol. 89, no. 17, pages 3484–3498. doi:[10.1177/0040517518811948](https://doi.org/10.1177/0040517518811948). Cited pages 23 and 48.
- [133] Oded Stein, Eitan Grinspun, and Keenan Crane, Jul. 2018. “Developability of triangle meshes”. *ACM Transactions on Graphics*, vol. 37, no. 4, pages 77:1–77:14. doi:[10.1145/3197517.3201303](https://doi.org/10.1145/3197517.3201303). Cited page 15.
- [134] Tomohiro Tachi, 2010. “Origamizing polyhedral surfaces”. *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 2, pages 298–311. Cited page 15.
- [135] Rasmus Tamstorf and Eitan Grinspun, Nov. 2013. “Discrete bending forces and their jacobians”. *Graphical Models*, vol. 75, no. 6, pages 362–370. doi:[10.1016/j.gmod.2013.07.001](https://doi.org/10.1016/j.gmod.2013.07.001). Cited page 32.
- [136] Chengcheng Tang, Pengbo Bo, Johannes Wallner, and Helmut Pottmann, Jan. 2016. “Interactive design of developable surfaces”. *ACM Transactions on Graphics*, vol. 35, no. 2, pages 12:1–12:12. doi:[10.1145/2832906](https://doi.org/10.1145/2832906). Cited page 10.
- [137] 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, pages 329–340. doi:[10.1145/3332165.3347949](https://doi.org/10.1145/3332165.3347949). Cited page 22.
- [138] Ye Tao, Yi-Chin Lee, Haolin Liu, Xiaoxiao Zhang, Jianxun Cui, Catherine Mondoa, Mahnoush Babaei, Jasio Santillan, Guanyun Wang, Danli Luo, et al., May 2021. “Morphing pasta and beyond”. *Science Advances*, vol. 7, no. 19. doi:[10.1145/3332165.3347949](https://doi.org/10.1145/3332165.3347949). Cited page 22.
- [139] Xavier Tellier, Cyril Douthe, Laurent Hauswirth, and Olivier Baverel, Apr. 2021. “Form-finding with isotropic linear weingarten surfaces”. In *Advances in Architectural Geometry 2020*, pages 18–37. Paris. URL: https://thinkshell.fr/wp-content/uploads/2019/10/AAG2020_02_Tellier.pdf. Cited page 12.
- [140] Skylar Tibbits, Jan. 2014. “4D printing: Multi-material shape change”. *Architectural Design*, vol. 84, no. 1, pages 116–121. doi:[10.1002/ad.1710](https://doi.org/10.1002/ad.1710). Cited pages 20 and 21.
- [141] Stephen Timoshenko, 1925. “Analysis of bi-metal thermostats”. *Journal of the Optical Society of America*. doi:[10.1364/JOSA.11.000233](https://doi.org/10.1364/JOSA.11.000233). Cited pages 19, 21, and 37.
- [142] Thibault Tricard, Vincent Tavernier, Cédric Zanni, Jonàs Martínez, Pierre-Alexandre

- Hugron, Fabrice Neyret, and Sylvain Lefebvre, Nov. 2020. “Freely orientable microstructures for designing deformable 3D prints”. *ACM Transactions on Graphics*, vol. 39, no. 6, pages 211:1–211:16. doi:[10.1145/3414685.3417790](https://doi.org/10.1145/3414685.3417790). Cited page 69.
- [143] Teunis van Manen, Shahram Janbaz, Mahya Ganjian, and Amir A. Zadpoor, 2020. “Kirigami-enabled self-folding origami”. *Materials Today*, vol. 32, pages 59–67. doi:<https://doi.org/10.1016/j.mattod.2019.08.001>. Cited page 19.
- [144] Teunis van Manen, Shahram Janbaz, and Amir A. Zadpoor, 2017. “Programming 2D/3D shape-shifting with hobbyist 3D printers”. *Materials Horizons*, vol. 4, pages 1064–1069. doi:[10.1039/C7MH00269F](https://doi.org/10.1039/C7MH00269F). Cited pages 20 and 21.
- [145] Teunis van Manen and Amir A. Zadpoor, 2021. “Theoretical stiffness limits of 4D printed self-folding metamaterials”. [arXiv:2106.05790](https://arxiv.org/abs/2106.05790). Cited pages 19 and 25.
- [146] Wim M. van Rees, Elisabetta A. Matsumoto, A. Sydney Gladman, Jennifer A. Lewis, and L. Mahadevan, 2018. “Mechanics of biomimetic 4D printed structures”. *Soft matter*, vol. 14, no. 43, pages 8771–8779. doi:[10.1039/C8SM00990B](https://doi.org/10.1039/C8SM00990B). Cited pages 20 and 21.
- [147] Wim M. van Rees, Etienne Vouga, and L. Mahadevan, 2017. “Growth patterns for shape-shifting elastic bilayers”. *Proceedings of the National Academy of Sciences*, vol. 114, no. 44, pages 11,597–11,602. doi:[10.1073/pnas.1709025114](https://doi.org/10.1073/pnas.1709025114). Cited pages 20, 34, 35, 49, and 51.
- [148] Amir Vaxman, Marcel Campen, Olga Diamanti, Daniele Panozzo, David Bommes, Klaus Hildebrandt, and Mirela Ben-Chen, 2016. “Directional Field Synthesis, Design, and Processing”. *Computer Graphics Forum*, vol. 35, pages 545–572. doi:[10.1111/cgf.12864](https://doi.org/10.1111/cgf.12864). Cited page 69.
- [149] Diederik Veenendaal and Philippe Block, 2012. “An overview and comparison of structural form finding methods for general networks”. *International Journal of Solids and Structures*, vol. 49, no. 26, pages 3741–3753. doi:[10.1016/j.ijsolstr.2012.08.008](https://doi.org/10.1016/j.ijsolstr.2012.08.008). Cited page 12.
- [150] Josh Vekhter, Jiacheng Zhuo, Luisa F Gil Fandino, Qixing Huang, and Etienne Vouga, Jul. 2019. “Weaving geodesic foliations”. *ACM Transactions on Graphics*, vol. 38, no. 4, pages 34:1–34:22. doi:[10.1145/3306346.3323043](https://doi.org/10.1145/3306346.3323043). Cited pages 16 and 69.
- [151] Pascal Volino, Nadia Magnenat-Thalmann, and François Faure, Sep. 2009. “A simple approach to nonlinear tensile stiffness for accurate cloth simulation”. *ACM Transactions on Graphics*, vol. 28, no. 4, pages 105:1–105:16. doi:[10.1145/1559755.1559762](https://doi.org/10.1145/1559755.1559762). Cited page 50.
- [152] 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, pages 1–12. doi:[10.1145/3306346.3323043](https://doi.org/10.1145/3306346.3323043).

[3290605.3300656](#). Cited pages 19, 20, and 21.

[153] 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, pages 623–635. doi:[10.1145/3242587.3242625](#). Cited pages 19, 20, and 21.

[154] Huamin Wang, James F. O’Brien, and Ravi Ramamoorthi, Jul. 2011. “Data-driven elastic models for cloth: Modeling and measurement”. *ACM Transactions on Graphics*, vol. 30, no. 4, pages 71:1–71:12. doi:[10.1145/2010324.1964966](#). Cited page 53.

[155] Stephanie Wang and Albert Chern, Jul. 2021. “Computing minimal surfaces with differential forms”. *ACM Transactions on Graphics*, vol. 40, no. 4, pages 113:1–113:14. doi:[10.1145/3450626.3459781](#). Cited page 11.

[156] 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, pages 6123–6132. doi:[10.1145/3025453.3026019](#). Cited page 22.

[157] Penelope Webb, Valentina Sumini, Amos Golan, and Hiroshi Ishii, 2019. “Auto-inflatables: Chemical inflation for pop-up fabrication”. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI EA ’19, pages 1–6. doi:[10.1145/3290607.3312860](#). Cited page 22.

[158] Guowei Yan, Wei Li, Ruigang Yang, and Huamin Wang, Dec. 2018. “Inexact descent methods for elastic parameter optimization”. *ACM Transactions on Graphics*, vol. 37, no. 6, pages 253:1–253:14. doi:[10.1145/3272127.3275021](#). Cited page 14.

[159] Jonas Zehnder, Espen Knoop, Moritz Bächer, and Bernhard Thomaszewski, Nov. 2017. “Metasilicone: Design and fabrication of composite silicone with desired mechanical properties”. *ACM Transactions on Graphics*, vol. 36, no. 6, pages 240:1–240:13. doi:[10.1145/3130800.3130881](#). Cited page 15.

[160] Kun Zhou, John Snyder, Baining Guo, and Heung-Yeung Shum, 2004. “Iso-charts: Stretch-driven mesh parameterization using spectral analysis”. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP ’04, pages 45–54. Nice. doi:[10.1145/1057432.1057439](#). Cited page 68.