C-shells: Deployable Gridshells with Curved Beams

QUENTIN BECKER, EPFL, Switzerland
SEIICHI SUZUKI, EPFL, Switzerland
YINGYING REN, EPFL, Switzerland
DAVIDE PELLIS, ISTI-CNR, Italy
JULIAN PANETTA, University of California, Davis, USA
MARK PAULY, EPFL, Switzerland

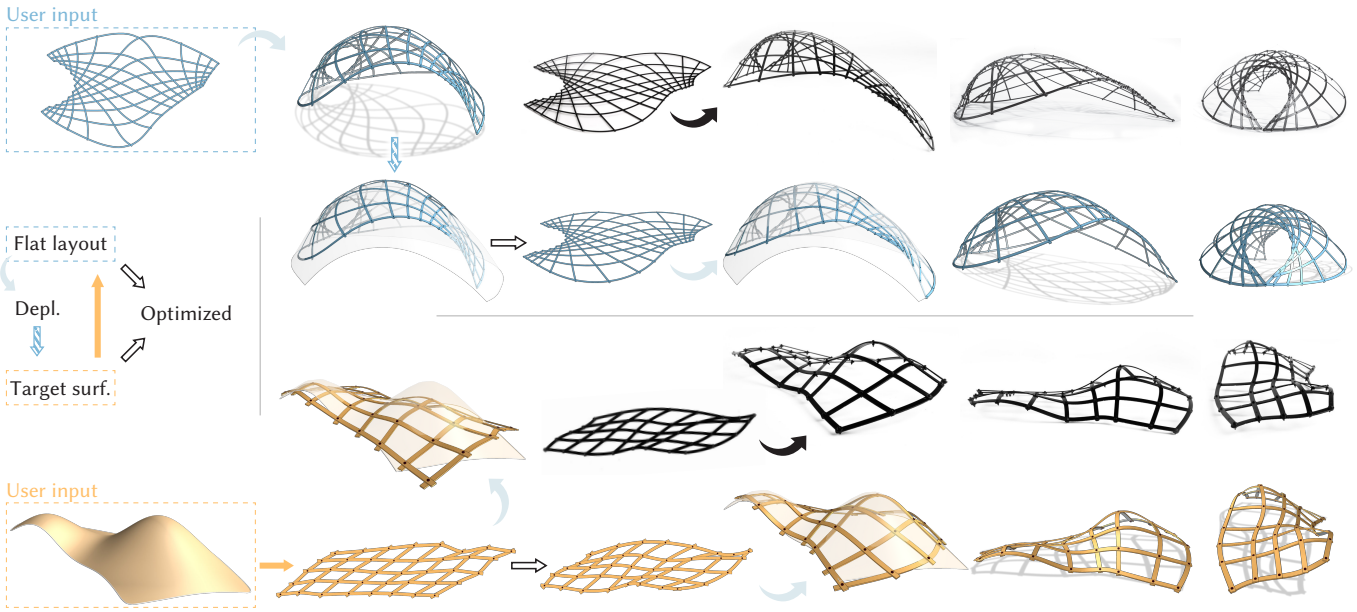


Fig. 1. Forward exploration (top) and inverse design (bottom) of C-shells. The target design surface is either provided directly by the user or created by deploying an initial beam layout. Subsequent design optimization then solves for the planar curved beam shapes such that the deployed linkage best approximates the target surface, while minimizing its total elastic energy. For inverse design, the initial linkage is computed using a novel flattening optimization. The three columns on the right show different views of the same deployed structure in each row. Fabricated prototypes are shown in black.

We introduce a computational pipeline for simulating and designing *C-shells*, a new class of planar-to-spatial deployable linkage structures. A C-shell is composed of curved flexible beams connected at rotational joints that can be assembled in a stress-free planar configuration. When actuated, the elastic beams deform and the assembly deploys towards the target 3D shape.

We propose two alternative computational design approaches for C-shells: (i) Forward exploration simulates the deployed shape from a planar beam layout provided by the user. Once a satisfactory overall shape is found, a subsequent design optimization adapts the beam geometry to reduce the elastic energy of the linkage while preserving the target shape. (ii) Inverse design is facilitated by a new geometric flattening method that takes a design surface as input and computes an initial layout of piecewise straight linkage beams. Our design optimization algorithm then calculates the smooth curved beams to best reproduce the target shape at minimal elastic energy.

We find that C-shells offer a rich space for design and show several studies that highlight new shape topologies that cannot be achieved with existing deployable linkage structures.

CCS Concepts: • **Computing methodologies** → **Modeling and simulation**.

Additional Key Words and Phrases: Computational design, deployable structures, physical simulations, fabrication

ACM Reference Format:
Quentin Becker, Seiichi Suzuki, Yingying Ren, Davide Pellis, Julian Panetta, and Mark Pauly. 2023. C-shells: Deployable Gridshells with Curved Beams. *ACM Trans. Graph.* 42, 6, Article 181 (December 2023), 17 pages. https://doi.org/10.1145/3618366

Authors' addresses: Quentin Becker, EPFL, Lausanne, Switzerland, quentin.becker@epfl.ch; Seiichi Suzuki, EPFL, Lausanne, Switzerland, seiichi.suzuki@epfl.ch; Yingying Ren, EPFL, Lausanne, Switzerland, yingying.ren@epfl.ch; Davide Pellis, ISTI-CNR, Pisa, Italy, davide.pellis@gmail.com; Julian Panetta, University of California, Davis, Davis, USA, jpanetta@ucdavis.edu; Mark Pauly, EPFL, Lausanne, Switzerland, mark.pauly@epfl.ch.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM 0730-0301/2023/12-ART181
https://doi.org/10.1145/3618366

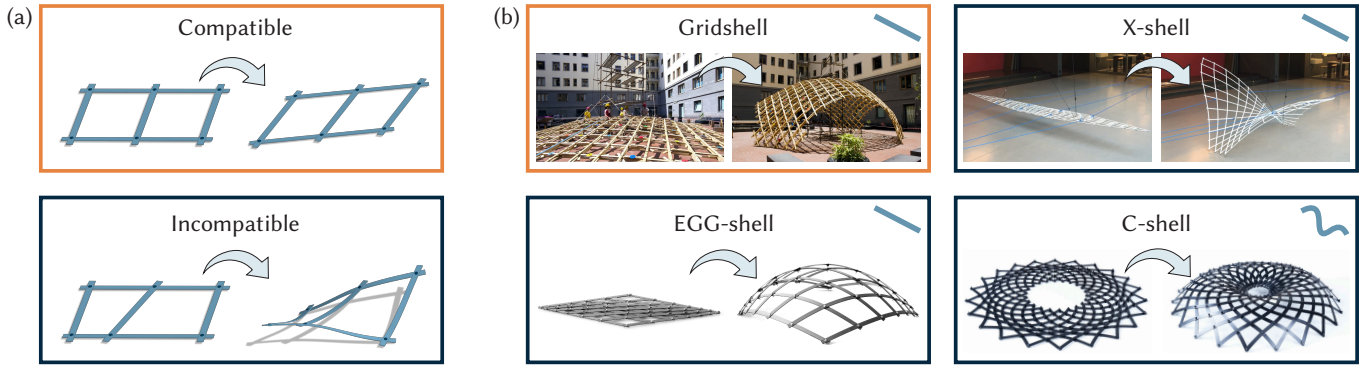


Fig. 2. Existing deployable beam structures (gridshell, X-shell, G-shell) assemble *straight* beams in a (near) planar configuration that can then be deployed into a desired 3D target surface. Traditional gridshells employ a compatible scissor linkage and are deployed by constraining the boundary. G-Shells and X-shells use an incompatible linkage and can be deployed using torque actuation at the joints. C-shells extend this methodology towards *curved* beams. Images taken from D'Amico et al. [2015] (Gridshell), Pillwein et al. [2020] (G-shell), and Panetta et al. [2019] (X-shell) are used with permission granted by the respective corresponding authors.

1 INTRODUCTION

Active bending builds on the controlled forming principle of inducing curvature on flat components using elastic deformations to structurally stabilize a material system [Lienhard 2014]. A key concept is the use of members with slender cross-sections and materials with high tensile strength that can sustain significant deformations as loads are applied.

Gridshells are a class of deployable structures that leverage active bending. A gridshell is composed of thin flexible beams that are connected at hinge joints to form a grid-like linkage. This leads to a complex deformation behavior where beams deform elastically while rotating around their joints.

Traditional gridshells use *straight* beams arranged in a *uniform* grid. Such scissor linkages expand and contract freely in-plane and can be deformed into curved shapes by imposing constraints on the boundary [Quinn and Gengnagel 2014]. More recent methods, such as X-shells [Panetta et al. 2019], G-shells [Soriano et al. 2019], and elastic geodesic grids [Pillwein et al. 2020; Pillwein and Musialski 2021], arrange straight beams in *non-uniform* grids. This nonuniformity leads to kinematic incompatibilities when the linkage is deployed, which forces the structure to buckle out of plane to reach an elastic equilibrium (see Figure 2). While often considered detrimental in engineering, buckling is essential in this context to achieve the desired transformation from planar assembly states to curved target surfaces.

A benefit of these nonuniform grids is that their target shapes are encoded directly in the material system rather than being determined by external forces like the boundary constraints imposed in traditional gridshells. This enables the shape transformation to be achieved via a variety of mechanisms including, e.g., intrinsic actuation via torque motors applied at the pivot joints.

In this paper, we introduce *C-shells*, a new class of deployable gridshells composed of optimized *curved* beams. Similar to X-shells and elastic geodesic grids, C-shells exploit spatial incompatibilities in the linkage to deploy the structure towards a 3D equilibrium state (see Figure 1). Compared to previous straight-beam structures, our generalization to curved beams offers a richer space for constructing freeform surface geometries and provides more flexibility for optimization (see Figure 21).

A key advantage of our approach compared to X-shells is that the fabrication state is guaranteed to be planar and free of any residual stress in the beams. This greatly simplifies assembly and enables new designs that cannot be achieved with X-shells (see also Figure 15). Elastic geodesic grids [Pillwein et al. 2020] also feature stress-free assembly states through the use of sliding notches, but these introduce weaknesses in the beams that can lead to material failure. On the other hand, curved beams require more complex fabrication tools compared to straight beams, which can be cut to length from stock material. Computationally, solving for curved rest shapes drastically increases the number of design variables and introduces new computational challenges to avoid invalid configurations when optimizing the beam layout.

To handle the additional complexity, we introduce a spline-based representation for the beams that reduces the free variables to the control points of the spline curves. Splines offer the additional advantage of implicitly regularizing the beam geometry towards smooth curves, which are generally preferred for reasons of visual aesthetics and structural performance.

Our spline-based representation enables interactively exploring variations of a flat beam layout by manipulating the joints positions (see Figure 10). For inverse design, when a target surface is provided as input, we propose a planarization algorithm that jointly optimizes for a planar linkage and its counterpart on the target surface so that both are kinematically as compatible as possible. Beam layouts produced in this way are further refined by a design optimization that solves for the rest shapes of the curved gridshell beams such that the deployed structure best approximates the target surface while minimizing the elastic energy of the structure. This requires tracking the equilibrium state of the deployed gridshell while its beam rest shapes and corresponding joint positions are optimized.

Our approach accurately simulates the physical deformation behavior of curved beams that can in principle deform to arbitrary

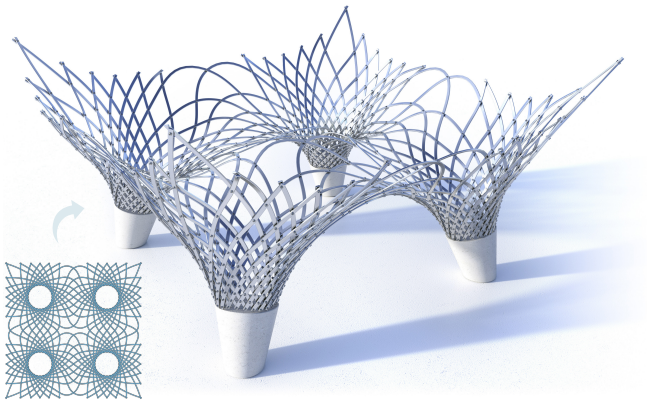


Fig. 3. A design study for a large-scale architectural application.

curves on the surface. While this requires more involved computations compared to the purely geometric approach of Pillwein et al. [2020], simulating the physics ensures accurate prediction of the deployed shapes, enables modeling of beams with arbitrary cross-section shapes and aspect ratios, and provides access to important structural aspects of the design, such as its total elastic energy and the distribution of stress within its beams upon deployment.

Contributions. Our main contributions can be summarized as:

- We introduce the concept of C-shells, deployable elastic gridshells composed of curved beams with guaranteed zero-energy planar assembly state.
- We present a numerical optimization to jointly solve for curved rest shapes of gridshell beams and their degree of actuation such that the deployed linkage has minimal elastic energy and best matches a given target geometry.
- We propose a forward design exploration method based on conformal maps.
- We present a flattening algorithm for inverse design to compute initial beam layouts suitable for optimization.

Using our computational tools, we explore new design typologies and fabricate several functional prototypes that demonstrate the predictive accuracy of our simulation and highlight the versatility of our design approach. The full source code and all design studies can be found at https://go.epfl.ch/c-shells.

2 RELATED WORK

Our work falls in the category of optimization-based design of planar-to-spatial deployable structures. Numerous material systems have been proposed in recent years that can be actuated from a planar fabrication state towards a curved 3D surface. Examples include tensioned membranes [Guseinov et al. 2017; Jourdan et al. 2022; Pérez et al. 2017], origami [Dudte et al. 2016], auxetic materials [Chen et al. 2021; Konaković-Luković et al. 2018], temperature-sensitive plates [Boley et al. 2019; Guseinov et al. 2020], and inflatables [Panetta et al. 2021].

We focus our discussion here on compliant gridshells that are composed of two families of elastic beams connected via rotational joints. We categorize such gridshells into two different classes (Extrinsic and Intrinsic) based on their deployment (Figure 2). For a discusson of other types of deployable scissor mechanisms, we refer to Ren et al. [2022].

Extrinsic deployment. Traditional elastic gridshells [Lienhard and Gengnagel 2018] feature straight elastic beams connected in a regular grid, allowing the structure to freely shear in-plane. Such systems can be analyzed and modeled using Chebyshev nets [Baek et al. 2018; Garg et al. 2014; Sageman-Furnas et al. 2019]. Deployment is achieved by constraining positions along the boundary, which forces the gridshell to deform out of plane to assume a state of minimum elastic energy.

Several works have introduced active bending structures composed of initially flat elements with in-plane rest curvature [LaMagna and Knippers 2017; Liu and Faisal 2022] or varying cross-section [Hafner and Bickel 2021]. Mhatre and colleagues [2021] proposed a type of deployable circular structure made from elastic beams with constant in-plane curvature and rectangular cross-section. Their study demonstrated the influence that beam geometry has on the deployed state. In contrast to our approach, beams are not explicitly coupled along their span by rotational joints; instead, the deployment process works by applying compressive forces to each beam end via a rotating mechanism. By optimizing tilings of spiral patterns cut into flat panels, FlexMaps [Malomo et al. 2018] tunes the sheet material's elasticity so that each panel deforms into a target patch of a 3D surface upon assembly. Laccone and colleagues [2021] investigate the use of this method in architectural applications, developing a computational design framework for curved bending-active structures and evaluating it through the construction of a large-scale prototype.

Intrinsic deployment. In contrast to the above methods, which rely on boundary constraints to deform the elastic structure towards its target state, another line of research designs structures with inbuilt deployment mechanisms based on geometric incompatibility. X-shells [Panetta et al. 2019] join straight beams in irregular grids that can be assembled in the plane and deployed into a 3D shape. This approach permits the designer to find a satisfactory 3D shape by iteratively editing the planar grid layout with interactive feedback on the deployed geometry. Once a satisfactory shape is found, the planar grid layout is optimized to reduce elastic energy in both the flat assembly configuration and the deployed states. A practical investigation of the design and materialization implications of X-shells was conducted by constructing a medium-scale pavilion [Isvoranu et al. 2019].

G-shells and elastic geodesic grids exploit the fact that thin lamellas are essentially constrained to follow geodesic curves on a surface due to their highly anisotropic bending stiffness. Soriano and colleagues [2019] develop a multi-objective optimization algorithm attempting to solve for a planar embedding of a given grid of geodesic curves. The flexibility of this design space can be increased by allowing the joint locations to slide along the beams, e.g., by introducing notches [Pillwein et al. 2020]; this not only expands the space of feasible surfaces but also enables stress-free assembly configurations. This later work introduces a purely geometric approach for simultaneously constructing a grid of geodesics on a curved

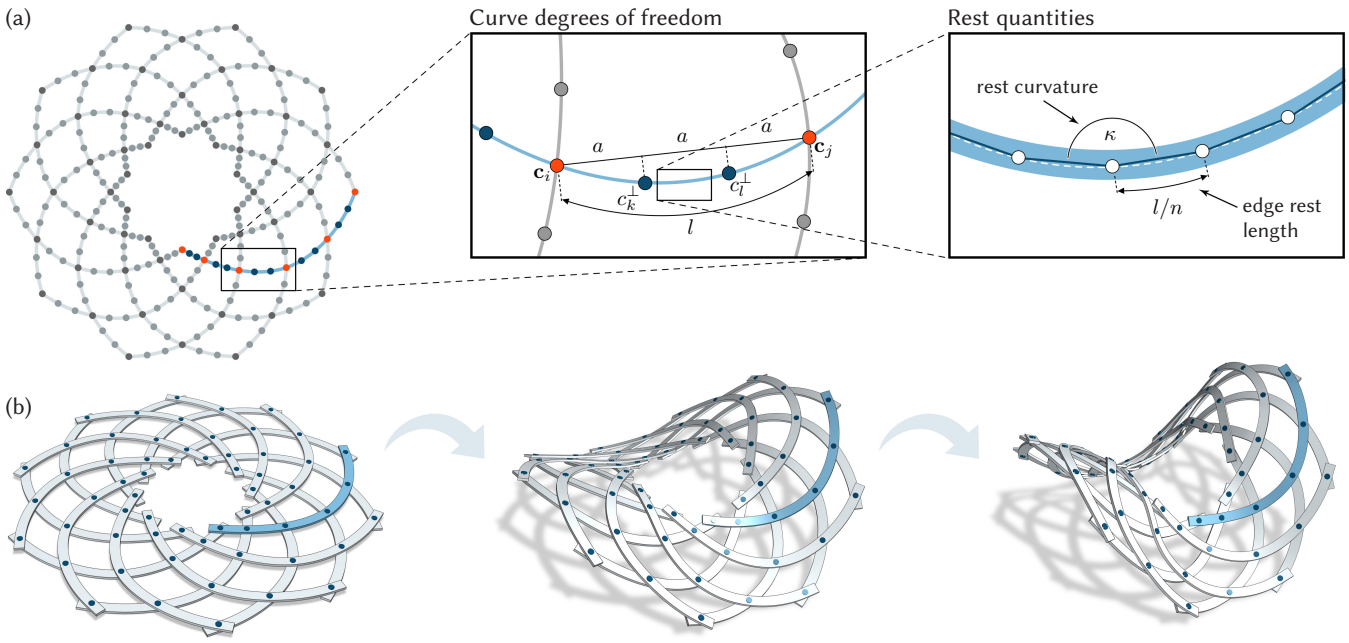


Fig. 4. C-shell representation. (a) The curve linkage is composed of interpolating splines that meet at joints $\mathbf{c}_i$. Intermediate interpolation points are evenly distributed along the chord joining consecutive joints and are constrained to move orthogonally to it according to scalar parameter $c_k^\perp$. Each curve is discretized into a polyline, from which we extract the discrete curvature κ. The arc length l is evenly distributed among edges' rest lengths. (b) The deployed equilibrium state is obtained by gradually increasing the deployment angle and minimizing the elastic energy stored in the curved beams.

surface patch and a corresponding planar grid into which it flattens via a single angle degree of freedom. In follow-up work, elastic geodesic grids were expanded to include multiple patches [Pillwein and Musialski 2021] and non-convex boundaries [Pillwein and Musialski 2021], though the latter involves actuation via boundary constraints.

Gridshell structures that follow asymptotic networks (A-shells) have been studied in Schling et al. [2018]. Since asymptotic lines have zero normal curvature, such networks can only exist on surfaces with non-positive Gaussian curvature. The absence of normal curvature enables A-shells to be constructed from straight planar strips with the thick cross-section axis oriented *normal* to the design surface. This alignment of the strong bending axis has benefits for the surface's load-bearing capacity. Schikore and colleagues [2021] propose a generalized framework for kinetic gridshells with straight beams and categorize them based on doubly ruled, geodesic, and asymptotic networks. A deployable system based on asymptotic gridshells has been presented in Schikore and Schling [2021].

Weaving. Also related to our work are recent computational design methods for 3D weaving. Traditional weaving interleaves straight ribbons that are commonly produced from natural materials such as bamboo [Ayres et al. 2018]. Singularities may be introduced in the weave pattern to control the weave's curvature [Ayres et al. 2021]. A computational inverse design approach for straight ribbons has been proposed by Vekhter and colleagues [2019]. Similar to G-shells and elastic geodesic grids, their approach traces ribbons along geodesics of the surface by extracting contours of a geodesic foliation. Ren and co-workers [2021] proposed a method that generalizes weaving towards curved ribbons, which provides more flexibility in terms of the topology of the weaving pattern and produces a smoother appearance of the final woven structure. While curved weaving design does not consider deployment from a planar state, our simulation framework for curved beams uses a similar simulation setup to the algorithm of Ren et al. [2021] as discussed in more detail below.

3 C-SHELL REPRESENTATION

The layout of an X-shell [Panetta et al. 2019] is uniquely defined by the positions of the linkage joints since all beams have straight rest geometry. In contrast, beams in a C-shell can in principle assume arbitrary planar curves. The deployment behavior then depends both on the joint positions and the curve geometries.

As in previous methods, we simulate the elastic deformation of our curve linkage using the Discrete Elastic Rods (DER) model [Bergou et al. 2010, 2008]. Specifically, our forward simulation builds upon the methods used for X-shells and curved weaving [Ren et al. 2021]. While the former uses a linkage of straight beams that can deploy via actuation, the latter utilizes a network of curved ribbons that is woven directly in its 3D state and does not deploy. Our approach unifies these methods and optimizes for deployable linkages of curved beams.

3.1 Spline Representation

To facilitate design optimization, we introduce an intermediate spline representation for the curved beams that offers two main

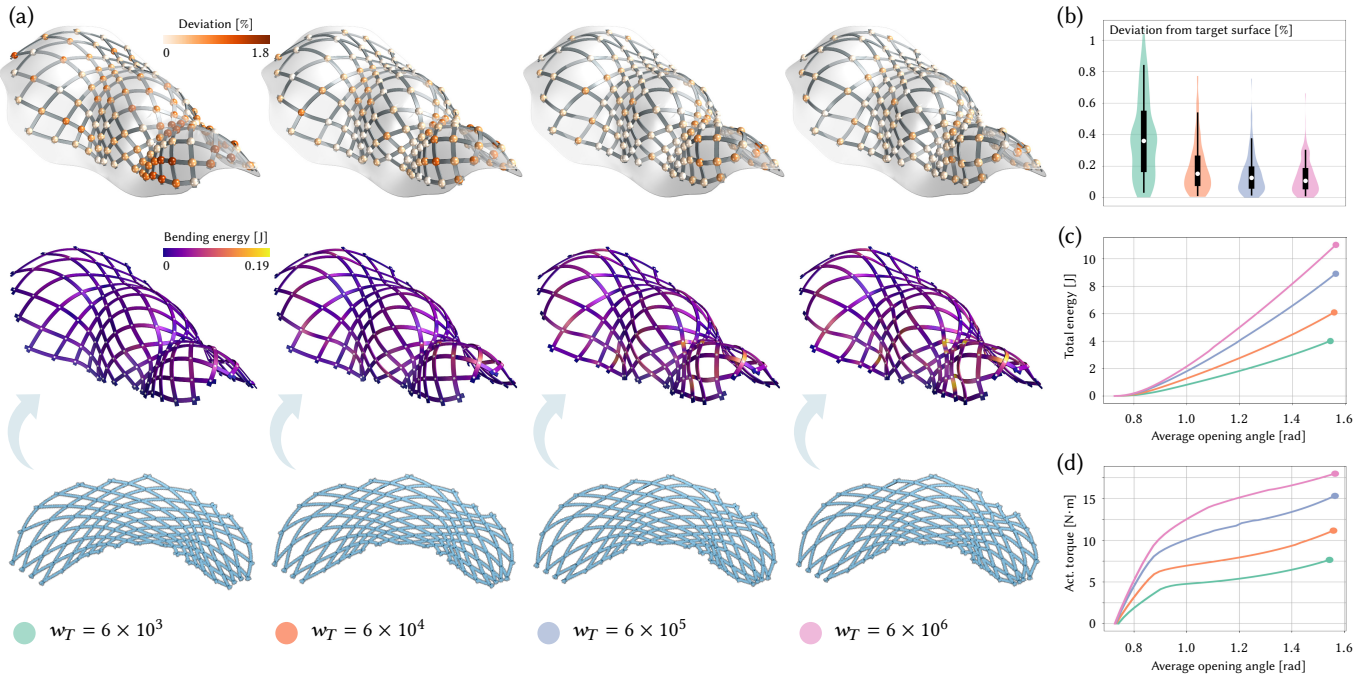


Fig. 5. (a) The surface fitting weight w_T controls the tradeoff between target fitting (top) and elastic energy (middle). Deviation to the target surface is reported as a percentage of the diagonal of the surface's bounding box. (b) Target surface fitting histograms measure the deviation from the target surface. White dots represent the medians, thick lines show the inter-quartiles, and thin lines indicate the 5th to 95th percentile range. (c) The total energy at a certain opening angle, (d) corresponding torque required to hold the structure.

advantages: (i) a significantly reduced number of optimization variables, and (ii) implicit regularization towards smooth curves.

We use natural cubic splines [Farin 2002] that interpolate the joint positions $\mathbf{c}_i \in \mathbb{R}^2$ to ensure that rod crossings occur exactly at these joints. For more control, we introduce additional interpolation points along the curve segment connecting each pair of joints. However, we constrain these *interior* control points to lie on evenly spaced lines perpendicular to the chord connecting the joints (see Figure 4). In other words, the interior control point positions are parameterized by perpendicular offset variables $c_k^\perp \in \mathbb{R}$, which combined with the joint positions, comprise our full vector of *design variables*:

$$\mathbf{q} = [\mathbf{c}_1, \ldots, \mathbf{c}_{n_{\text{joints}}}, c_1^\perp, \ldots, c_{n_{\text{ip}}}^\perp,]^\mathsf{T} \in \mathbb{R}^{n_q}.$$

These parameters uniquely define a C-shell's planar rest state.

3.2 Rod Representation

We convert each spline curve in the linkage into a DER centerline by sampling at uniform intervals to obtain a polyline; we typically subdivide the arc connecting each pair of joints into ten edges.

Rest state variables. The rest state of each discrete rod in the converted linkage is defined by the lengths of each edge and a discrete curvature value at each vertex computed from the turning angle of consecutive edges of the polyline (see Figure 4). Beams are assumed to be untwisted in their planar rest states. These rest lengths and rest curvatures are grouped into the vector of DER rest-state quantities, $\mathbf{p} \in \mathbb{R}^{n_p}$.

We note that our spline sampling conversion defines a differentiable mapping from the design variables $\mathbf{q}$ to DER rest-state quantities $\mathbf{p}(\mathbf{q})$, enabling us to solve the design optimization problem (Section 5) over the reduced design variables $\mathbf{q}$ instead of directly optimizing rest vertex positions of the DER representation.

Simulation variables. We simulate the deformation of the curve linkage by minimizing the DER stretching, bending, and twisting energies [Bergou et al. 2010, 2008] stored in each beam. The beams are coupled at the joints using the rotational joint model of Ren and co-workers [2021], which enforces positional and orientation constraints using a nonlinear change of variables. We refer to those papers for further details.

All DER centerline position and material frame angle variables left unconstrained by the joints are aggregated along with the joint state variables (position, orientation, opening angle, and crossing edge lengths) into a vector $\mathbf{x} \in \mathbb{R}^{n_x}$ uniquely defining the C-shell's deformed state.

4 FORWARD SIMULATION

As proposed by Panetta et al. [2019], we drive the deployment of the curve linkage by imposing an average opening angle across all rotational joints. This constraint effectively applies the same torque to every joint, and allows individual joint angles to adapt to minimize the global elastic energy during deployment.

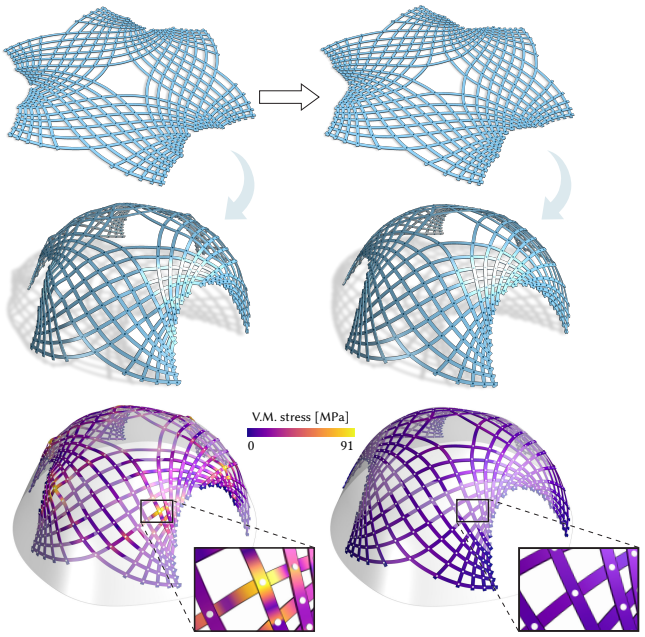


Fig. 6. Manual design of a C-shell linkage often leads to high stresses in the deployed shape (left). Our optimization reduces stress concentrations without significantly affecting the deployed shape, which helps avoid material failure during deployment (right). Here the maximum von Mises stress is reduced by 78%.

4.1 Deployed Equilibrium State

To simulate the 3D deployed state corresponding to an imposed average opening angle of $\bar{\alpha}^{\text{tgt}}$, we solve the discrete minimization problem:

$$\mathbf{x}^*_{\text{3D}}(\mathbf{p}, \bar{\alpha}^{\text{tgt}}) := \underset{\mathbf{x}}{\text{argmin}} \; \frac{1}{YV_0} E(\mathbf{x}, \mathbf{p}) + T(\mathbf{x}) \qquad (1)$$
$$\text{s.t.} \quad \bar{\alpha}(\mathbf{x}) = \bar{\alpha}^{\text{tgt}},$$

where E is the elastic energy summed over all DERs in the linkage, T is the surface-attraction energy defined later in Equation (4), and $\bar{\alpha}(\mathbf{x})$ computes the average opening angle from the simulation variables. The material's Young's modulus Y and a reference C-shell volume V_0 serve scaling the energy term. We follow the rigid-motion-pinning strategy in Ren and colleagues [2021], adding T here with a small weight (1.0×10^{-5}) to factor out the global rigid motion of the C-shell and ensure favorable alignment with a target input surface.

As we gradually increase $\bar{\alpha}^{\text{tgt}}$, the structure buckles out of its flat state and realizes a 3D shape. To accelerate this incremental deployment simulation, we employ the same continuation approach used in Panetta and colleagues [2019], constructing a first-order approximation $\mathbf{x}^*_{\text{3D}}(\mathbf{p}, \bar{\alpha}^{\text{tgt}} + \Delta\bar{\alpha}) = \mathbf{x}^*_{\text{3D}}(\mathbf{p}, \bar{\alpha}^{\text{tgt}}) + \frac{\partial \mathbf{x}^*_{\text{3D}}}{\partial \bar{\alpha}^{\text{tgt}}} \Delta\bar{\alpha} + O(\Delta\bar{\alpha}^2)$ for the next equilibrium using sensitivity analysis.

One significant deviation from the X-shells simulation framework was needed regarding the constraint $\bar{\alpha}(\mathbf{x}) = \bar{\alpha}^{\text{tgt}}$. Panetta et al. [2019] impose this as a general linear equality constraint, effectively solving a linearized KKT system in each Newton step. However, for efficiency, and to simplify detection of unstable equilibria, they solve this indefinite system using block elimination so that a Cholesky factorization of only the energy's Hessian H is needed. The assumption underlying this approach is that H is positive definite in the neighborhood of a solution to (1), i.e., near stable equilibria. Unfortunately, this assumption does not always hold: at a *constrained* minimizer, H still can have a direction of negative curvature (in our case, along the deployment path). In these situations, the X-shells solver would apply unnecessary Hessian modifications in every Newton step and refuse to converge, falsely detecting that the solution it is converging to as an unstable equilibrium. This situation is rare for X-shells away from the initial onset of buckling, but we have found it prevalent for C-shells.

We propose a different approach for enforcing the equality constraint that resolves this issue: we apply a linear change of variables that exposes $\bar{\alpha}(\mathbf{x}) := \bar{\alpha}_{\mathbf{x}}$ as an explicit state variable in $\mathbf{x}$. This transforms the dense linear equality constraint into a single variable pin constraint $\bar{\alpha}_{\mathbf{x}} = \bar{\alpha}^{\text{tgt}}$ that can be applied by eliminating $\bar{\alpha}_{\mathbf{x}}$ from the

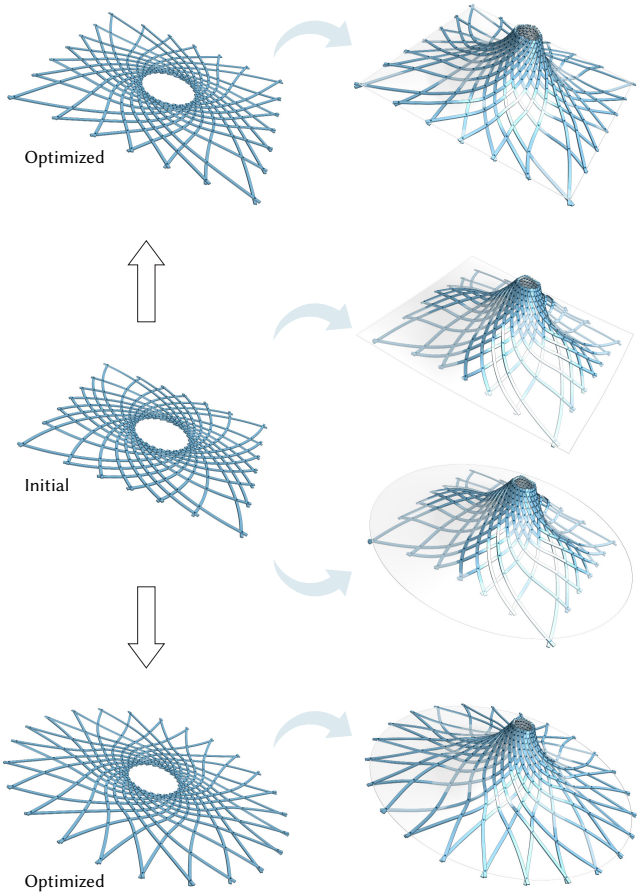


Fig. 7. Controlling target fitting. Specifying 200 times higher weights for the square (top) and circle (bottom) boundary points allows optimizing the initial design (middle) to accurately reproduce the desired target shapes.

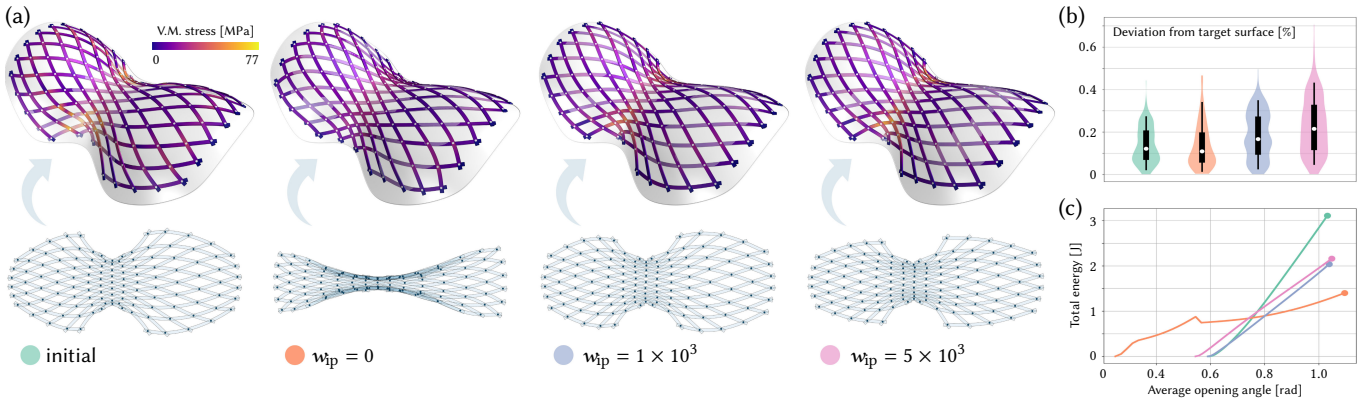


Fig. 8. Effect of Laplacian regularization on the design optimization. (a) Stress distribution in the deployed state (top) and planar linkage layout. Self-intersections are clearly noticeable without regularization. (b) Target fitting histograms. White dots represent the medians, thick lines show the inter-quartiles, and thin lines indicate the 5th to 95th percentile range. Deviation is reported relative to the surface bounding box diagonal. (c) Total energy of deployed state.

optimization problem (removing one row and column of H before solving the Newton system). This smaller Hessian *is* guaranteed to be positive definite in a neighborhood of a solution to (1), and the optimization reliably converges. Care must be taken to define this linear change of variables in a way that preserves the sparsity of H; we present the details of how we achieve this in the supplemental material.

4.2 Stress Computation

Stresses in the deployed state are computed by adapting the approach from Megaro and colleagues [2017] as explained in the supplemental.

5 DESIGN OPTIMIZATION

Forward simulation allows predicting the deployed state of a C-shell given its planar layout. The design optimization algorithm aims to reduce the total elastic energy of the deployed C-shell while best matching a given target design (Figure 6). We discuss in Section 6 how to obtain the initial beam layout through either forward exploration or inverse design process.

5.1 Objective Formulation

To optimize the beam layout, we propose an efficient algorithm for adapting the curve linkage using derivatives of the forward simulation. We jointly optimize the spline design variables *and the target average opening angle*, as opposed to the fixed average opening angle that was used in X-shells. Optimizing the deployment in this way adds an important degree of flexibility that can improve the target surface approximation and reduce the elastic energy of the deployed state.

Our C-shell design optimization seeks to minimize the objective:

$$J(\mathbf{x}, \mathbf{q}, \mathbf{p}) := \frac{1}{E_0} E(\mathbf{x}) + T(\mathbf{x}) + R(\mathbf{q}, \mathbf{p}), \tag{2}$$

where R is a regularization term defined below, and E_0 is the initial value of the elastic energy of the deployed state. Specifically, we compute the spline parameters and deployment angles solving:

$$\begin{aligned} &\underset{\mathbf{q}, \bar{\alpha}^{\text{tgt}}}{\text{argmin}}\, J(\mathbf{x}^*_{\text{3D}}(\mathbf{p}(\mathbf{q}), \bar{\alpha}^{\text{tgt}}), \mathbf{q}, \mathbf{p}(\mathbf{q})) \\ &\text{s.t.} \quad \alpha_{\min}(\mathbf{q}) \geq \epsilon. \end{aligned} \tag{3}$$

Note that to avoid self-intersections of the planar rest state, the minimum opening angle $\alpha_{\min}(\mathbf{q})$ in this state is constrained to be greater than some positive angle ϵ. We use a smoothed approximation of the minimum given by the negation of the LogSumExp (LSE) function defined as $\text{LSE}(\boldsymbol{\alpha}, s) := s \log(\sum_i \exp(-\alpha_i / s))$, where $\boldsymbol{\alpha}$ holds all the opening angles and $s>0$ trades off between smoothness and adherence to the exact minimum. We fix $s=0.01$ in our experiments.

Target surface fitting term. The term T in (2) penalizes deviations from a given surface. To prevent the rod linkage from shrinking, we add a term measuring the distance from the joints to user-defined target positions $\mathbf{x}_t$. The full term is given by

$$T(\mathbf{x}) := \frac{w_T}{2l_0^2} \left(\|\mathbf{x} - P_{\text{surf}}(\mathbf{x})\|^2_{W_{\text{surf}}} + \|\mathbf{x} - \mathbf{x}_t\|^2_{W_t} \right), \tag{4}$$

where w_T controls the importance given to the target-fitting term and P_{surf} projects points onto their closest points on the target surface while leaving the non-positional degrees of freedom unchanged.

We normalize T using the length of the deployed linkage bounding box diagonal, l_0. Diagonal matrices W_{surf} and W_t extract the joint positions from the simulation variables $\mathbf{x}$, and allow adjusting the importance of different regions of the target surface. The two weight matrices are set so that their coefficients are non-negative and sum to one, for w_T to effectively control the importance given to the surface fitting term.

Figure 5 illustrates the trade off between deployed elastic energy and target fitting. We run our design optimization on the same initial design for varying values of fitting weight w_T. Increasing w_T helps to better fit the prescribed target surface at the expense of larger elastic energy in the deployed state and greater torque required to maintain the deployment. As the histograms in Figure 5 indicate, the target-fitting objective saturates at $w_T = 6 \times 10^5$.

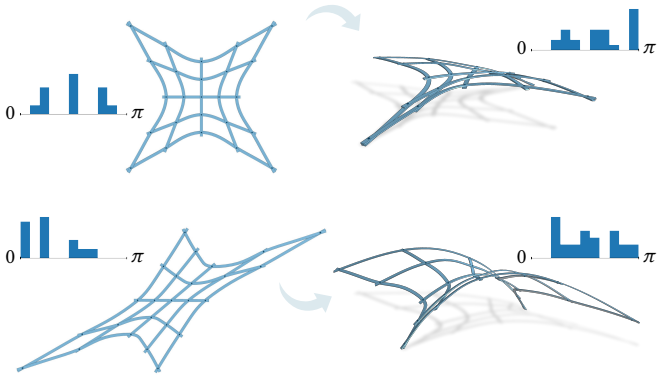


Fig. 9. Impact of the angle distribution in the flat state regarding achievable deployed shapes. Having angles spanning a larger range in the flat state (top row) does not allow deploying the linkage as much as when angles are more consistently closed or opened (bottom row). The histograms show the distribution of opening angles for each state of each design.

Figure 7 shows how the target-fitting term can be spatially adapted using the weight matrices. The initial design poorly fits the square and circular boundaries of the target surfaces. After assigning greater importance to these regions by selecting larger entries in W_t corresponding to these boundary joints, the optimized designs approximate the respective boundaries well.

Regularization term. Our design parameterization allows joints to move freely in plane, which can result in extreme curvature despite the implicit regularization provided by our spline representation. Furthermore, the orthogonal offsets $c_k^{\perp}$ may cause the rest state to have self-intersections. To mitigate the risk of such collisions, we introduce a regularization composed of two terms:

$$R(\mathbf{q}, \mathbf{p}) := w_{\text{ip}} R_{\text{ip}}(\mathbf{q}) + w_{\kappa} R_{\kappa}(\mathbf{p}). \tag{5}$$

The first term R_{ip} is a fairness term that pushes each interpolated point towards the average of its neighbors. The second term R_{κ} penalizes the deviation of the rest curvature at a vertex of a discrete rod to the average curvature of its two neighboring vertices along that rod; this encourages beams to assume circular arcs. Scalar weights w_{ip} and w_{κ} control the importance given to each of the two terms in Equation (5).

Figure 8 illustrates the effect of regularization on the design optimization. We run our design optimization starting from the same initial design for different regularization weights w_{ip}. Increasing w_{ip} helps prevent self-intersections and straightens the beams at the expense of higher stress, larger total energy, and increased deviation of the deployed structure from the target surface. For further details, please refer to the supplemental material.

6 DESIGN CONCEPTION

The design optimization presented in the previous section requires a suitable initial estimate of the planar rest state of the C-shell. We propose two strategies for designing such initial linkages, *forward exploration* and *inverse design*, illustrated in Figure 1. In the forward approach, a designer directly draws and edits curves to generate a planar C-shell linkage that is deployed to create an initial target design. For inverse design, we propose a novel planarization algorithm that takes a target surface as input and computes an initial beam layout. In both cases, our design optimization algorithm is subsequently applied to find optimal shapes of the curved beams that reduce the elastic energy in the deployed configuration while best preserving the target surface. This target surface can then be further edited through smooth deformations to explore nearby C-shell designs.

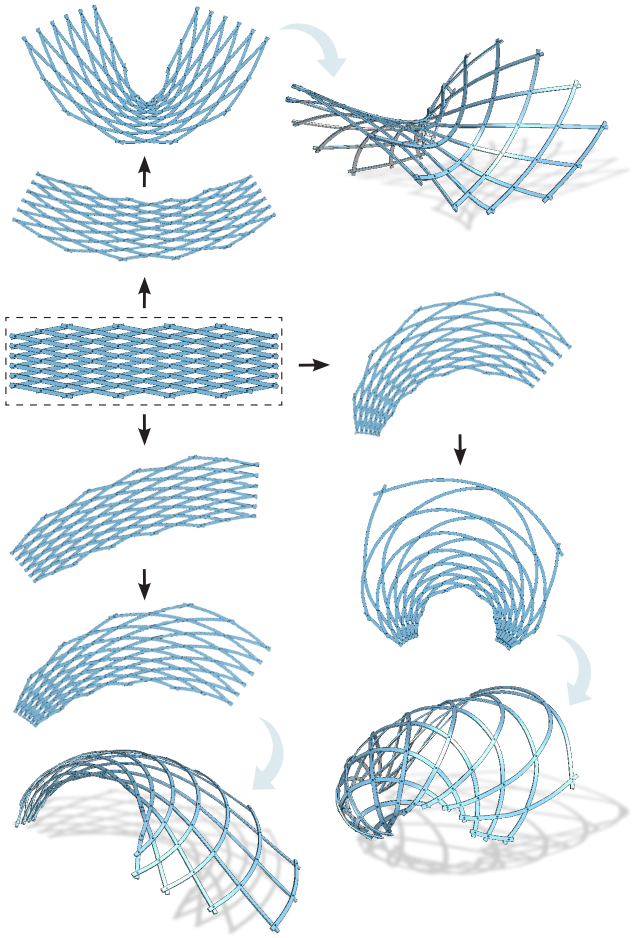


Fig. 10. Editing beam geometry using conformal deformations in the plane offers great flexibility to adapt an initial regular layout. These deformations preserve the angle distribution at the joints to retain a large deployment range.

6.1 Forward Exploration

To explore the space of C-shells, the user can in principle draw an arbitrary planar network of intersecting curves as an initial layout. However, arbitrarily drawn curves tend to intersect in a large range of angles, which can significantly restrict deployability. When certain joints have initial angles close to π or 0, the region around these joints cannot be further deployed (by either increasing

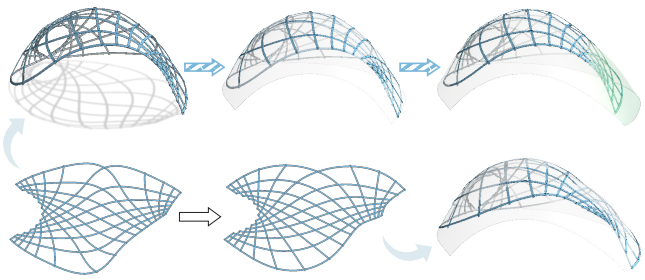


Fig. 11. C-shell editing. Once an overall satisfactory beam layout and target shape have been found, the C-shell can be further edited by smoothly deforming the target surface. The design optimization then adapts the planar layout accordingly. Regions of larger edits are indicated in green.

or decreasing the average angle, respectively) without inverting the layout. As illustrated in Figure 9, a wide range of initial angles limits the ability of the linkage to transform and achieve interesting 3D shapes.

This observation motivates the following design strategy: Starting from a regular grid of straight beams, the designer modifies the linkage interactively by applying injective planar conformal maps in the form of Möbius transforms. This warps the layout into one with curved beams, but preserves the narrow range of joint angles, since a conformal map is angle-preserving. However, inter-joint distances are changed, so the warped linkage will no longer remain compatible and thus deploy into a 3D state.

As illustrated in Figure 10, conformal maps offer great versatility for transforming initially straight beam layouts towards curved beam shapes. These kinds of transformations and the corresponding design flexibility are enabled by the additional degrees of freedom of C-shells compared to straight beam linkages. Combined with fast visual feedback of the deployed shape from our deployment simulation, such edits enable effective design exploration. Crucially, no residual stresses are imposed on the planar beams under such modifications, which guarantees that a valid physical assembly exists for any user edits.

Several examples we show are designed with this strategy (see Figure 5, 8, 18, and five examples from Figure 21). Additionally, ring-shaped layouts can be designed by parameterizing families of curves and copying them around a circle (see Figure 4, 7, 15, 6, 16, 20). Lastly, different layouts can be composed by cutting and reconnecting their boundary curves as shown in Figure 3.

Editing. Initial planar layouts created with the forward process are improved by running our design optimization discussed in Section 5. We can further edit the design by directly manipulating the target design surface through standard deformation tools. When running the design optimization with the modified design surface, the target fitting term (4) will pull the deployed C-shell towards the new target and optimize the curved beam shapes accordingly. If edits to the target surface are sufficiently small, the design optimization is generally successful in mapping the C-shell to the new surface. For larger edits, this process can be broken down into a series of smaller steps to ensure the design optimization is able to track the new target surface without getting stuck in an unfavorable local minimum. Figure 11 shows an example of such an editing operation.

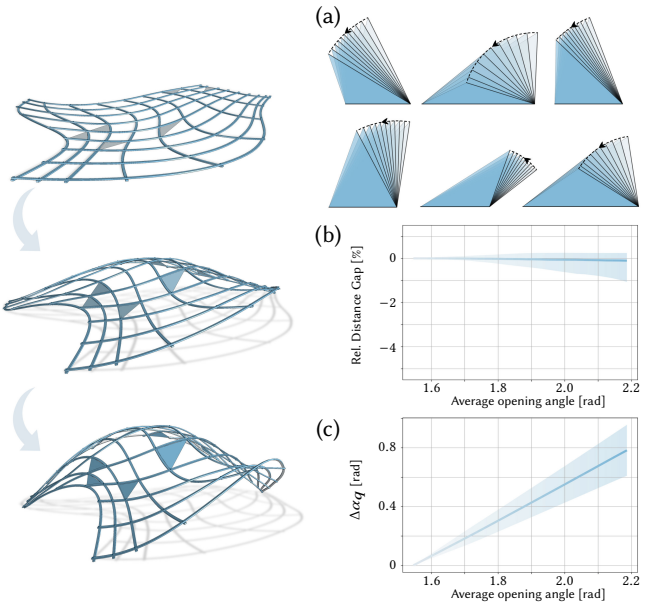


Fig. 12. C-shell deployment kinematics. (a) Isolated triangles connecting neighboring joints in the linkage reveal that one vertex follows an approximately circular trajectory when the opposite edge is kept fixed during deployment. Inter-joints distances remain approximately constant with a maximum relative change of 1.8% during deployment. (b) median and 5th to 95th percentile range of relative distances. (c) mean and standard deviation of the opening angle increment as the linkage deploys.

6.2 Planarization

While forward exploration is suitable to *discover* interesting target shapes, inverse design is more appropriate when a target surface is already provided as input. For an input regular linkage topology, our planarization algorithm jointly optimizes the joint positions $\mathbf{c}$ in the flat state and $\bar{\mathbf{c}} := (S(u_1, v_1), S(u_2, v_2), \ldots)$ on the surface S, so that a C-shell connecting the joints $\mathbf{c}$ in the flat state deploys to align these joints with the corresponding locations $\bar{\mathbf{c}}$ on the target surface.

Our algorithm assumes a smooth surface patch $S : [0, 1]^2 \to \mathbb{R}^3$ as input. To reduce computational complexity, we simplify each curve as a piece-wise straight polyline connecting joints, so that each rod segment is reduced to a single edge. The smooth curved beams will then be created during the later design optimization step from this initial coarse layout.

Planarization seeks a minimum of a weighted sum of terms:

$$\begin{aligned}\mathbf{c}^*, \mathbf{u}^*, \mathbf{v}^* := \underset{\mathbf{c},\mathbf{u},\mathbf{v}}{\operatorname{argmin}}\ & \mathcal{E}_{\text{len}} + \mathcal{E}_{\alpha} + \mathcal{E}_{\text{eq}} + \mathcal{E}_{\text{bnd}} \\ \text{s.t. } & C_{\text{quads}} \geq 0, u_i \in [0, 1], v_i \in [0, 1], \end{aligned} \tag{6}$$

where we dropped variable dependencies for conciseness. These terms are motivated from observations regarding the kinematics of the deployment illustrated in Figure 12.

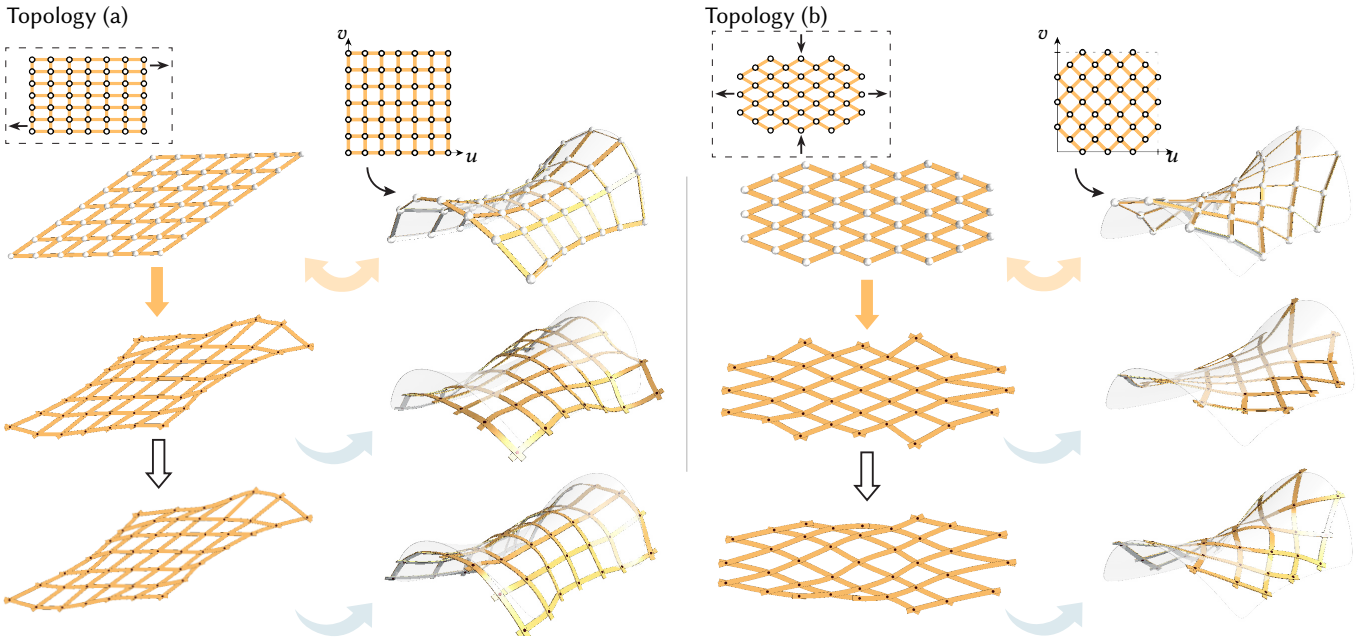


Fig. 13. Our planarization algorithm applied to different linkage layouts. Topology (a) follows the uv-isolines to trace the target joint positions. Topology (b) follows uv-isolines rotated by $\pi/4$. The flat layouts are initialized by deploying the regular grids as indicated in the dashed boxes and globally scaling the beams to match the measured lengths in the target state. For each example, planarization computes the layout shown in the middle row as an initialization to the design optimization shown in the bottom row.

Joints on the surface are initialized by following uv-isolines. Joint positions in the planar state are initialized by shearing a square grid, where the length of each side of the sheared grid matches the average of the deployed polyline lengths (see also Figure 14). We choose a shearing angle of approximately 60° in our experiments.

We now present the different terms and constraints involved in the optimization.

Segment length preservation. In general, the Euclidean distance between two neighboring joints changes during deployment because linkage beams bend and twist. However, as Figure 12 illustrates, this variation is very small. In the example shown, more than 90% of the inter-joint distances vary by less than 1%. This motivates our length-preserving term defined as

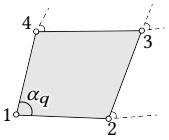
$$\mathcal{E}_{\text{len}}(\mathbf{c},\mathbf{u},\mathbf{v}) := \frac{w_{\text{len}}}{2\,\bar{l}_0^{\,2}} \sum_c \sum_i \left(\left\| \mathbf{c}_{c,i+1} - \mathbf{c}_{c,i} \right\| - \left\| S(u_{c,i+1}, v_{c,i+1}) - S(u_{c,i}, v_{c,i}) \right\| \right)^2, \quad (7)$$

where $\mathbf{c}_{c,i}$ is the i-th joint along curve c, $\bar{l}_0$ is the initial average polyline length in the deployed state, and w_{len} controls the importance given to this term. We use a default value of $w_{\text{len}} = 5 \times 10^3$ in our experiments.

Opening angle increment spread. As illustrated in Figure 9, and confirmed through experiments with many designs, joint angles generally either open or close globally during deployment. Specifically, we observed that the variance of opening angle increments $\Delta\alpha_q$ during deployment is typically low (see also Figure 12). We capture this behavior with a term that minimizes this variance around each joint as

$$\mathcal{E}_\alpha(\mathbf{c},\mathbf{u},\mathbf{v}) := \frac{w_\alpha}{2} \mathbb{V}\text{ar}_q[\Delta\alpha_q]. \quad (8)$$

To define $\Delta\alpha_q$, we number the vertices of each quadrilateral q in the linkage from 1 to 4 consistently and define the opening angle increment as



$$\Delta\alpha_q := \angle(\bar{\mathbf{c}}_{q,2} - \bar{\mathbf{c}}_{q,1}, \bar{\mathbf{c}}_{q,4} - \bar{\mathbf{c}}_{q,1}) - \angle(\mathbf{c}_{q,2} - \mathbf{c}_{q,1}, \mathbf{c}_{q,4} - \mathbf{c}_{q,1}), \quad (9)$$

where $\angle(\mathbf{a},\mathbf{b})$ is the signed angle between two vectors $\mathbf{a}$ and $\mathbf{b}$. The weight is set to $w_\alpha = 5$ by default in our experiments.

Equilibrium. The linkage defined by the flat joints must be at equilibrium under torque actuation when the joints are moved to their locations on the surface. We introduce a term that minimizes the residual force magnitude required to hold the joints at their target positions.

We build a simpler physical model of the linkage by describing each curve by a single DER: each rod segment is now an edge, and each DER's vertices are given by the joints along the rod. The rest quantities of each DER is given by the joints positions in the flat state. The material frame angles are determined so that they minimize the total elastic energy in the deformed configuration. The average angle in the simplified model is given by $\bar{\alpha}(\bar{\mathbf{c}})$ so that the

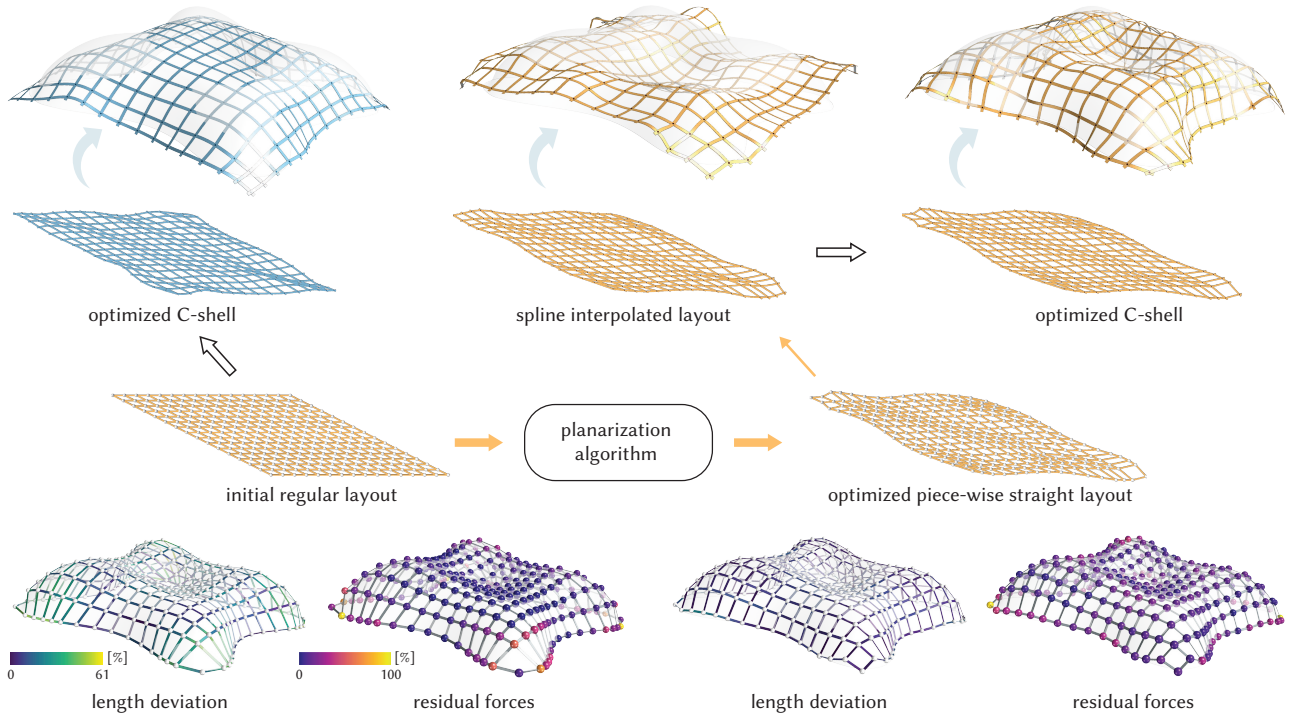


Fig. 14. Our planarization is essential to get a good initialization for design optimization. When starting from a regular grid (left), the design optimization fails to capture the bumps of the target surface adequately. The flattening algorithm computes an initial layout that does not accurately deploy towards the target (middle), but that can be successfully optimized to capture the surface detail (right). The bottom row indicates how length deviation and residual forces are reduced during the flattening optimization.

force balance equation residuals are given by

$$\mathcal{E}_{\text{eq}}(\mathbf{c},\mathbf{u},\mathbf{v}) := \frac{w_{\text{eq}}}{2(YA_0)^2}\left\|\frac{\partial E}{\partial \bar{\mathbf{c}}} - \lambda^* \frac{\partial \bar{\alpha}}{\partial \bar{\mathbf{c}}}\right\|^2, \tag{10}$$

where we scale forces with the material's Young modulus Y and cross sections area A_0. λ^* is the torque magnitude that minimizes the residuals. It is computed analytically at each optimization step. w_{eq} is the weight associated to this term, and is set to 10^{-2} by default in our experiments. A more detailed treatment of this equilibrium term can be found in the supplemental material.

Soft boundary pinning. To prevent the whole linkage from shrinking, we push the boundary joints $(u_{b,i}, v_{b,i})$ to some input positions $\bar{\mathbf{c}}_{b,i}^{(\text{tar})}$ using soft constraint

$$\mathcal{E}_{\text{bnd}}(\mathbf{u},\mathbf{v}) := \frac{w_{\text{bnd}}}{2\ \bar{l}_0^{\,2}} \sum_i \left\| S(u_{b,i}, v_{b,i}) - \bar{\mathbf{c}}_{b,i}^{(\text{tar})} \right\|^2 \tag{11}$$

where w_{bnd} is the regularization weight and defaults to $w_{\text{bnd}} = 5$ in our experiments.

Self-intersection-free quadrilaterals. We ensure that no quadrilateral in the flat layout is self-intersecting. For each quadrilateral in the linkage, we enforce that at least one of its two triangulations has two positively oriented triangles. More details are provided in the supplemental material.

Figure 13 shows the result of the planarization optimization for two different grid layouts applied to the same target surface. A more complex inverse design example is shown in Figure 14 for the lilium tower. As illustrated, direct design optimization starting from a regular initial layout does not yield a satisfactory result as the optimization gets stuck in an unfavorable local minimum. With our inverse flattening approach, we find a much better initial layout that can be successfully optimized to capture the details of the design surface. Note that this model can neither be achieved with X-shells nor with G-shells (refer to Figure 19 in Pillwein and Musialski [2021]).

7 NUMERICAL SOLVER

We solve both the design optimization problem in Equation (3) and the planarization problem in Equation (6) using a trust-region active-set method (SLQP) [Nocedal and Wright 2006] with a BFGS Hessian approximation as implemented by Knitro [Waltz and Nocedal 2004]. Gradients must be provided to the optimization algorithm. We describe next how we treat the terms in the problems.

Design optimization. The minimization of the design optimization objective is done with respect to the curves' design parameters. We first calculate analytical gradients of the design objective

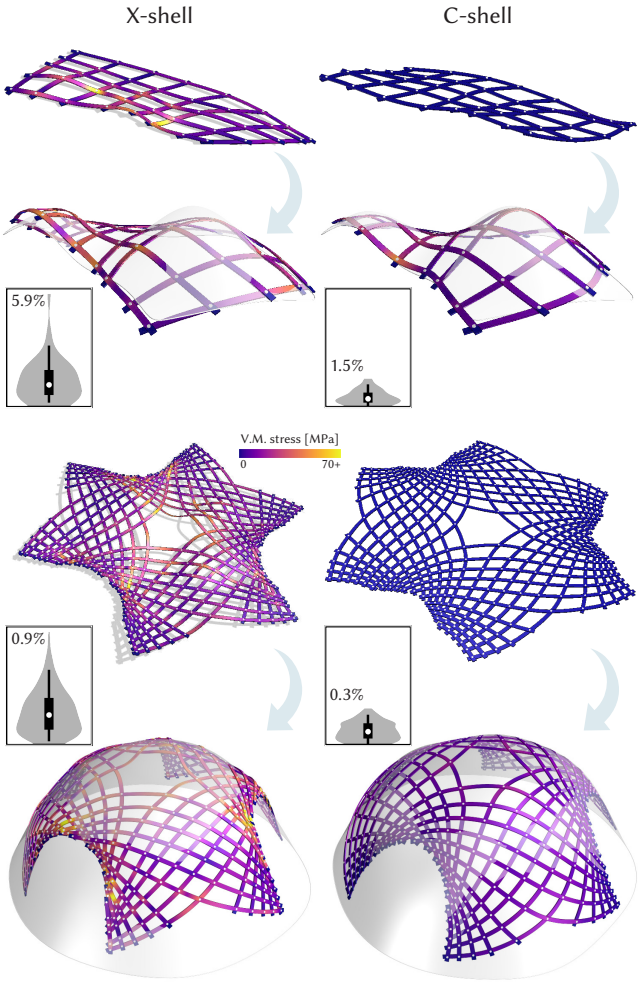


Fig. 15. X-shells vs. C-shells. The rest quantities and target shape have been chosen such that both deployed gridshells approximate the target well. The crucial difference is that X-shells have significant stress in the rest state, which no longer remains planar. The von Mises scale is cropped to the flexural yield stress of our acrylic material. Any value above that threshold indicates a high probability of local plastic deformation or fracture. Additionally, the C-shell models all outperform their X-shell counterparts in terms of deviation to the target surface. The deviation distribution is reported as violin plots where the maximum value (expressed as a percentage of the bounding box diagonal) is indicated. In each plot, the white dot represents the median value, the thick line the interquartile range, and the thin line the 5th to 95th percentile range.

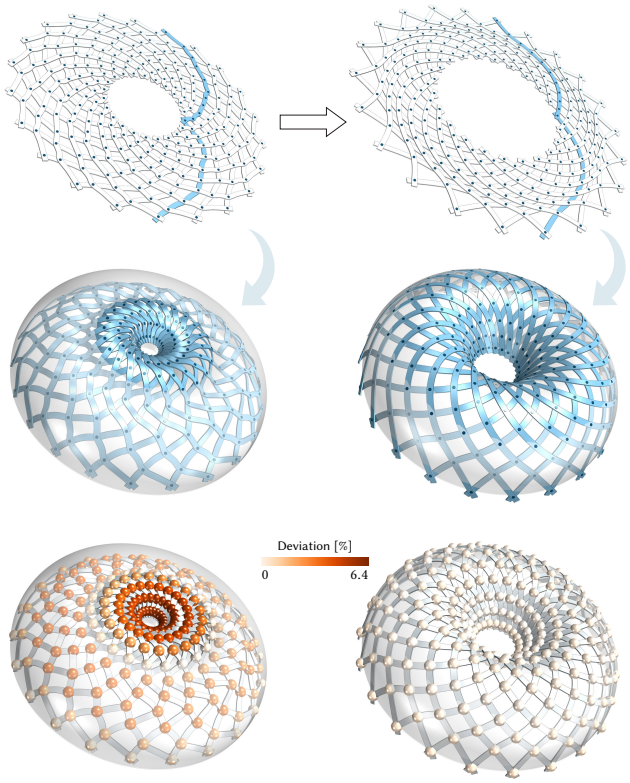


Fig. 16. Exploiting symmetry. For a symmetric target surface such as the torus, we explicitly enforce beams to be identical in each linkage family using a reduced representation highlighted in the planar layouts. The initial design (left) is optimized to best approximate the target (right). The deviation to the target surface is indicated as a percentage of the bounding box diagonal at each joint.

$\bar{J}(\mathbf{q},\mathbf{p},\bar{\alpha}^{\text{tgt}}) := J(\mathbf{x}^*_{\text{3D}}(\mathbf{p},\bar{\alpha}^{\text{tgt}}),\mathbf{q},\mathbf{p})$ with respect to the DER rest-state quantities $\mathbf{p}$ and deployment angle $\bar{\alpha}^{\text{tgt}}$ using adjoint sensitivity analysis as detailed in the supplemental material. We then backpropagate these analytical gradients through the spline-to-DER conversion using reverse-mode automatic differentiation implementation of the PyTorch library [Paszke et al. 2019] to obtain gradients with respect to our design variables, $\frac{\partial \bar{J}(\mathbf{q},\mathbf{p}(\mathbf{q}),\bar{\alpha}^{\text{tgt}})}{\partial \mathbf{q}} = \frac{\partial \bar{J}}{\partial \mathbf{p}}\frac{\partial \mathbf{p}}{\partial \mathbf{q}} + \frac{\partial \bar{J}}{\partial \mathbf{q}}$.

Planarization. The flat and target joints positions are jointly optimized, and gradients with respect to both are needed. The gradients of the terms $\mathcal{E}_{\text{len}}$, $\mathcal{E}_{\alpha}$, $\mathcal{E}_{\text{bnd}}$ with respect to the flat $\mathbf{c}$ and target $\bar{\mathbf{c}}$ joints positions are computed using automatic differentiation implemented in PyTorch. The gradient of $\mathcal{E}_{\text{eq}}$ with respect to the joints positions in both the flat and target configurations is obtained through adjoint sensitivity analysis as explained in the supplemental material. We further backpropagate the gradients to the uv-parameters using the chain rule e.g., $\frac{\partial \mathcal{E}_{\text{len}}(\mathbf{c},\bar{\mathbf{c}}(\mathbf{u},\mathbf{v}))}{\partial \mathbf{u}} = \frac{\partial \mathcal{E}_{\text{len}}}{\partial \bar{\mathbf{c}}}\frac{\partial S}{\partial \mathbf{u}}$. Surface derivatives $\frac{\partial S}{\partial \mathbf{u}}$ and $\frac{\partial S}{\partial \mathbf{v}}$ are computed using the geomdl Python library [Bingol and Krishnamurthy 2019]. The Jacobian of the constraints with respect to the flat joints positions $\mathbf{c}$ is obtained through automatic differentiation. Our planarization typically takes between 1 and 10 minutes to converge on the examples we show.

8 RESULTS

We evaluate our optimization-based framework in a series of design studies that illustrate the unique properties of C-shells. For example, our curved beams eliminate stress in the planar fabrication state (Figure 15), reduce stresses in the deployed state (Figure 6), and

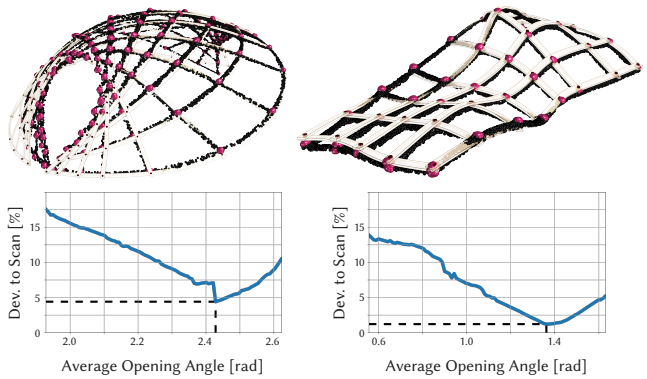


Fig. 17. The simulated models show good aggreement with the 3D scans of the fabricated prototypes. The physical models are deployed by hand, and held in place by either attaching strings to constrain the distance between pairs of joints (left), or by fastening the screws at the joints (right). Since we cannot measure accurately the average opening angle of the scanned model, we sweep the average opening angle of the simulated model and find the best match. The corresponding digital model is overlaid on the scan. The deviations of the joints positions to the scanned joints (shown in red) are respectively 4.4% (left model) 1.2% (right model) relative to the diagonal of the bounding box of the scanned model.

allow for more flexibility in terms of linkage topology (Figures 3 and 21). This enables designs that cannot be realized with any of the existing straight-beam gridshells models. Specifically, using the code of Panetta et al. [2019], we could not obtain geometrically satisfactory and physically realizable result for the design surfaces we show in this paper. Moreover, the algorithms presented in Panetta et al. [2019] only feature design optimization, but no inverse approach. Our new inverse flattening algorithm finds a suitable initialization for design optimization by exploiting the additional degrees of freedom that C-shells offer by allowing beams to be curved. A comparable method is currently not available for X-shells.

Fabrication. All our physical prototypes have been fabricated by first laser cutting the curved beams from modified acrylic sheets of 2 mm thickness (see Figures 1 and 20). We assemble the two families of beams using screws that act as rotational joint axes. We slightly widen the beam cross-sections around the joints to compensate for the hole required for the screw connection.

A crucial advantage compared to X-shells is the stress-free rest state that allows us to simply lay out the entire network of curves in the plane instead of forcing straight beams to bend and pass through the prescribed joint locations. As illustrated in Figure 15, this leads to high stress in the straight beams of the X-shells, which already buckle in the assembly state. This greatly complicates assembly and can easily lead to material failure. At the same time, we do not need to use notches as proposed in Pillwein et al. [2020], which weakens the beams and can complicate the deployment process.

We deploy our models simply by expanding the linkage by hand. While this deployment process deviates from the canonical torque actuation at the joints used in our simulation, the resulting 3D configuration closely resembles the simulated prediction since shape is robustly encoded in the linkage (see Figure 17).

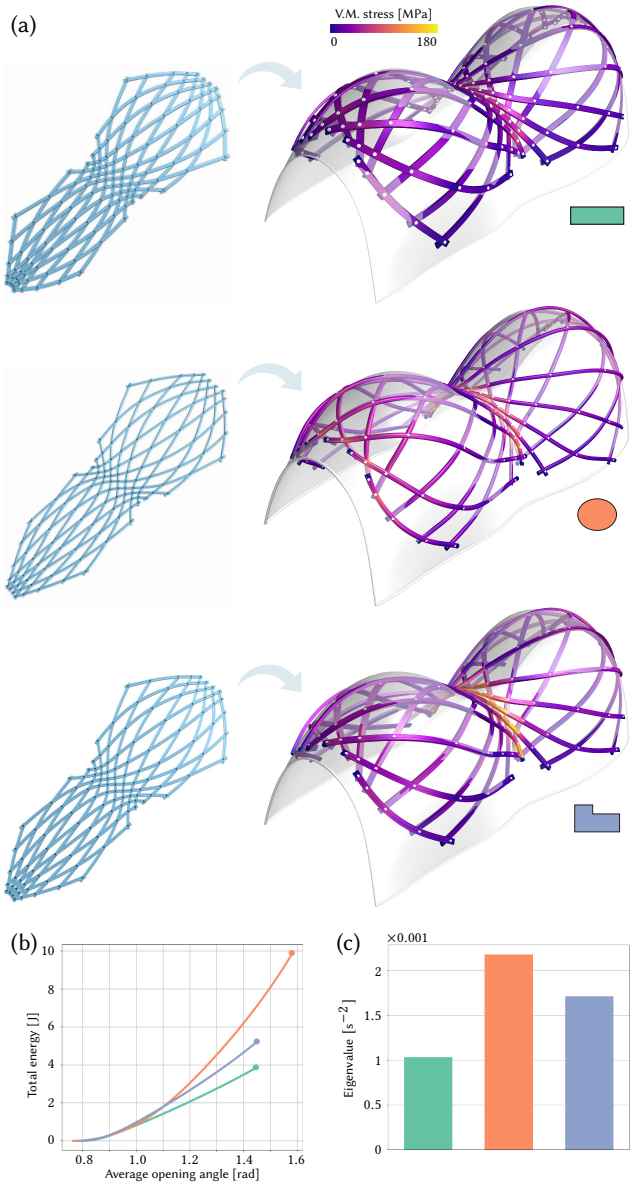


Fig. 18. Effect of varying the cross-section. We optimize an initial linkage towards the same target surface using three different cross-sections, all sharing the same area. The resulting linkages have a comparable median deviation to the target surface: respectively 0.16%, 0.17%, 0.15% of the bounding box diagonal. (a) The flat and deployed optimized designs for varying cross-sections. The von Mises stress distribution in the deployed state heavily depends on the cross-section shape. (b) Linkages with different cross-sections require different amounts of energy to be deployed. (c) The eigenvalue associated to the most compliant deformation mode of each deployed linkage. A higher value indicates a stiffer model.

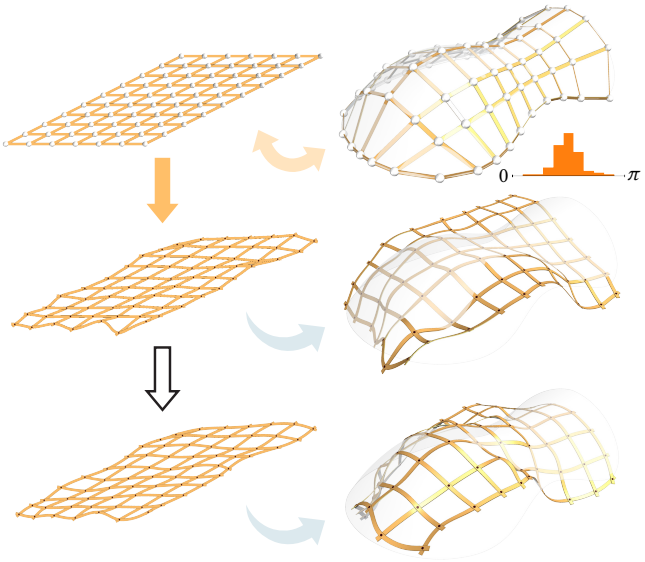


Fig. 19. A failure case for our inverse design pipeline. Due to the smoothness of the boundary, the opening angles must span a large spectrum as shown at the top, which is incompatible with highly curved deployed states (see Figure 9).

Symmetric designs. Figure 16 shows a design where we explicitly impose identical beam shapes in each linkage family. We implement this design constraint using a reduced representation where only the control points of one beam per family are unknowns in the optimization. Identical beams can simplify fabrication, in particular for larger-scale designs, and naturally lead to symmetric deployed states when employed in combination with a regular linkage topology.

Beam cross-section. In Figure 18 we show the effect of varying the beam cross-section, which affects the deformation behavior and stiffness of the linkage. We evaluate the stiffness of the deployed models by exposing their most compliant deformation. This deformation mode $\mathbf{v}$ around the deployed state $\mathbf{x}^*_{3D}$ can be found by solving the generalized eigenvalue problem $H\mathbf{v} = \mu M\mathbf{v}$, where H is the Hessian of the total elastic energy evaluated at $\mathbf{x}^*_{3D}$, M is the mass matrix of the linkage, and μ is the eigenvalue associated with the deformation mode $\mathbf{v}$.

In particular, this experiment shows that adjusting the cross-section affects the stress concentration and the stiffness of the deployed state. The rectangular cross-section has lower internal stress and is more compliant than the ellipsoidal cross-section. On the other hand, the ellipsoidal cross-section has lower internal stress while being stiffer than the L-shaped cross-section.

We report timings and metrics for some examples from the paper in Table 1.

9 LIMITATIONS AND FUTURE WORK

Fabrication complexity. A fundamental limitation of C-shells compared to previous methods that use straight beams is the increased fabrication complexity. The curved beams need to be cut or shaped,

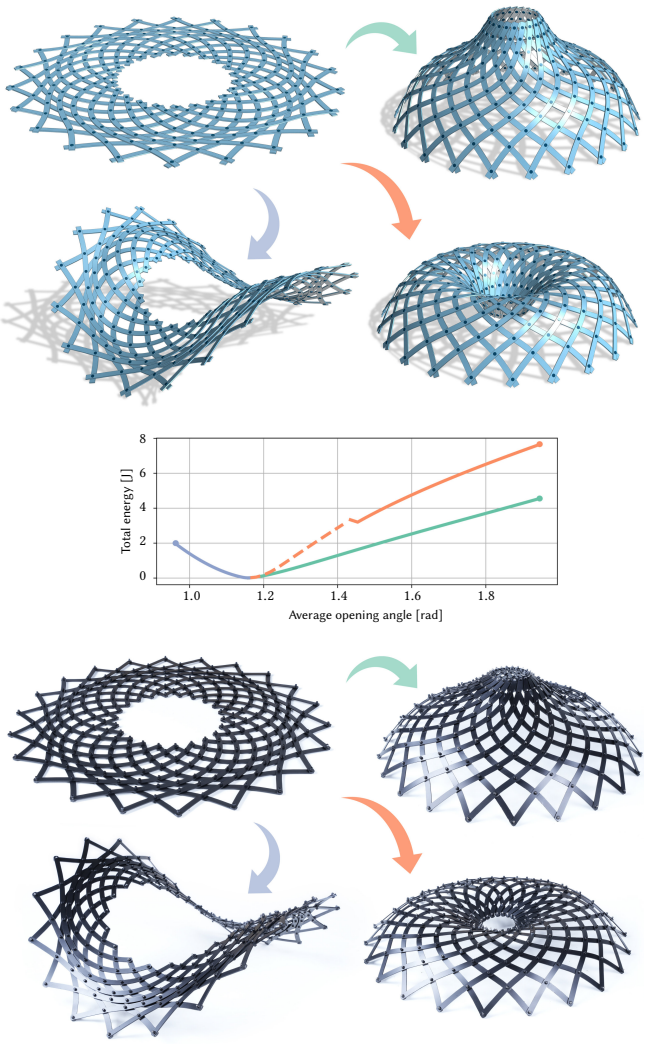


Fig. 20. Multi-deployment. This model has been optimized towards a toroidal shape (orange), but can also deploy towards a cone-like shape (green) and a saddle-like surface (purple). The energy plot indicates that the toroidal shape has higher energy than the cone shape. Hence, additional external forces are required to push the deployment towards the torus as indicated by the dashed line. Fabricated models at the bottom confirm that this behavior is also observed on physical C-shells.

which makes the use of natural materials difficult, as these often have strongly anisotropic behavior (e.g. wood). Curved beams also lead to more waste when cut from sheets. We currently try to minimize such waste by manually packing the computed beams as tightly as possible. An interesting question for future work could be how to directly optimize the beam shapes for tight fits.

Self-intersections. Currently, we do not explicitly prevent self-intersections of beams during design optimization or deployment. Our regularization aims at reducing the risk of self-intersections, yet we cannot guarantee an intersection-free final result. Manual

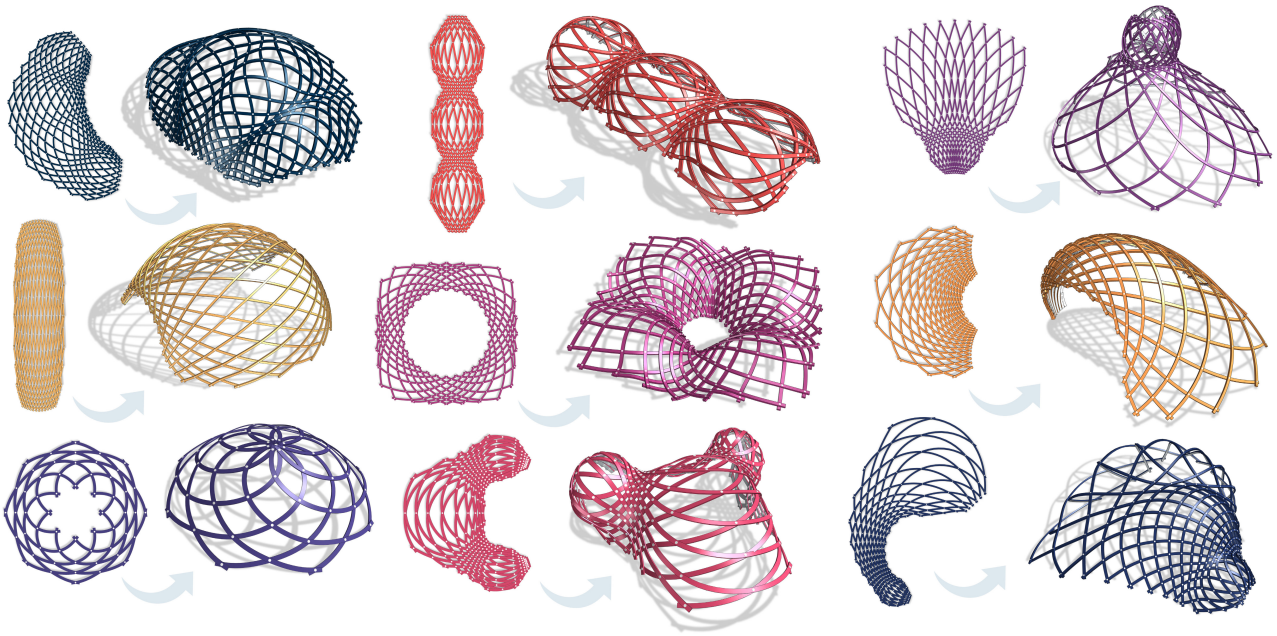


Fig. 21. C-shells offer increased flexibility in terms of linkage topology, which leads to a significantly richer shape space compared to deployable linkages with straight beams.

Table 1. Evaluation of the deployment simulation and the design optimization on several models. *#J* is the number of joints, *#R* is the number of rods. Deployment (*Sim. Time*) and design optimization (*Opt. Time*) are in min:sec and are measured on a Linux workstation with a 64-Core AMD Ryzen Threadripper 3990X Processor and 128GB of RAM. We execute our design optimization on 12 threads. *Tgt. Fit.* reports the maximum deviation of the optimized deployed rod linkage to the target surface as a percentage of the maximum deviation of the initial deployed linkage to the target surface. *Final En.* gives the total elastic energy of the optimized deployed linkage as a percentage of the total elastic energy of the initial deployed linkage. *Final Obj.* is the final objective value expressed as a percentage of the initial objective value.

Models (Figures)	#J	#R	n_q	Sim. Time	Opt. Time	Tgt. Fit. (%)	Final En. (%)	Final Obj. (%)
Dome (1, 11)	142	30	986	00:16	49:20	16.91	109.76	46.61
3Bumps, w_T=6×10^3 (5)	127	26	866	00:22	07:26	67.68	43.22	47.05
3Bumps, w_T=6×10^4 (5)	127	26	866	00:23	03:12	48.66	65.76	51.25
3Bumps, w_T=6×10^5 (5)	127	26	866	00:24	03:51	47.58	96.16	27.84
3Bumps, w_T=6×10^6 (5)	127	26	866	00:24	03:31	41.69	118.85	16.36
Square Boundary (7)	312	48	1824	00:34	1:05:42	19.18	88.03	15.00
Circular Boundary (7)	312	48	1824	00:34	5:25:04	8.28	105.11	0.79
Bat, w_{ip}=0 (8)	127	26	710	00:20	15:16	104.95	43.90	65.72
Bat, w_{ip}=1×10^3 (8)	127	26	710	00:14	07:26	110.64	65.49	71.61
Bat, w_{ip}=5×10^3 (8)	127	26	710	00:16	16:51	159.22	69.18	52.65
Hexagon (15)	453	66	2694	01:05	2:22:25	25.69	16.91	20.57
Torus (16)	242	44	62	00:42	19:40	10.71	14.20	5.26
2Bumps, rect. (18)	127	26	920	00:15	31:36	39.54	76.94	35.95
2Bumps, ellipsoidal (18)	127	26	920	00:08	33:34	44.28	97.95	32.08
2Bumps, L-shaped (18)	127	26	920	00:11	34:56	56.07	92.14	44.60

intervention is required if self-intersections are encountered, such as increasing ϵ and the regularization weights. While in all our examples, self-intersections did not pose difficulties, it would be interesting for future work to directly incorporate collision prevention into the optimization and thus reduce the design complexity for the user.

Material design. The cross-section shapes and material properties of the beams are currently exposed as user parameters in our optimization. Since these quantities significantly affect the deformation behavior of the linkage, it will be interesting to integrate them as variables in our optimization.

Topology design. In our current system, we expect the linkage topology to be specified as part of the design input and only optimize the continuous parameters of the model. Incorporating topology optimization into the pipeline could further simplify design and uncover new high-performance classes of designs.

Planarizeable target surfaces. Currently, our planarization algorithm expects an uv-parameterized surface patch with four opposing and distinct borders as input. However, some surfaces may not allow such a representation: see the torus in Figure 16. As illustrated in Figure 19, the planarization algorithm can fail to produce a suitable initialization for the subsequent design optimization. In this example, the alignment to the curved boundary poses particular difficulties that the optimization cannot resolve successfully.

Multi-stability. In our design shown in Figure 20, we observed an interesting phenomenon: despite being optimized to deploy towards a toroidal shape, the planar fabrication state can also be actuated towards two additional geometrically distinct stable shapes. This behavior emerged unintentionally, yet if it can be controlled during design optimization, such multi-stability would open up fascinating possibilities for shape-shifting and reconfigurable structures. Further research is needed to better understand when multiple stable states exist in C-shells and how to potentially optimize for multiple target states in the same design.

10 CONCLUSION

C-shells are a new type of deployable gridshell, enriching the catalog of bending-active linkage systems. While more involved in terms of numerical optimization and physical fabrication, the increased flexibility of curved beam linkages enables new opportunities for design. Novel grid topologies facilitate the exploration of deployed shape geometries that are not realizable with existing methods. At the same time, construction is simplified by ensuring a stress-free planar assembly state. With these unique benefits, C-shells have the potential for applications at a wide range of scales, from micro-engineered compliant robots to large-scale architectural constructions.

ACKNOWLEDGMENTS

We thank Filip Goč for his advices and his precious help in the realization of the prototypes. We are grateful to Michele Vidulis and Uday Kusupati for helping with the assembly of the prototypes. We also thank the anonymous reviewers for their valuable feedback. This research was supported by the Swiss National Science Foundation (Grant FNS 514543 / CF 1156).

REFERENCES

Phil Ayres, Alison Grace Martin, and Mateusz Zwierzycki. 2018. Beyond the Basket Case: a principled approach to the modelling of Kagome weave patterns for the fabrication of interlaced lattice structures using straight strips. In *Advances in Architectural Geometry (AAG)*.

Phil Ayres, Ji You-Wen, Jack Young, and Alison Grace Martin. 2021. Meshing with Kagome Singularities: Topology adjustment for representing weaves with double curvature. In *Advances in Architectural Geometry*. Ponts Chaussées, 188–207.

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* 115, 1 (2018), 75–80. https://doi.org/10.1073/pnas.1713841115 arXiv:https://www.pnas.org/doi/pdf/10.1073/pnas.1713841115

Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2010. Discrete viscous threads. *ACM Transactions on graphics (TOG)* 29, 4 (2010), 1–10.

Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete elastic rods. In *ACM SIGGRAPH 2008 papers*. 1–12.

Onur Rauf Bingol and Adarsh Krishnamurthy. 2019. NURBS-Python: An open-source object-oriented NURBS modeling framework in Python. *SoftwareX* 9 (2019), 85–94.

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* 116, 42 (2019), 20856–20862.

Tian Chen, Julian Panetta, Max Schnaubelt, and Mark Pauly. 2021. Bistable auxetic surface structures. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–9.

B D'Amico, A Kermani, H Zhang, A Pugnale, S Colabella, and S Pone. 2015. Timber gridshells: Numerical simulation, design and construction of a full scale structure. In *Structures*, Vol. 3. Elsevier, 227–235.

Levi H. Dudte, Etienne Vouga, Tomohiro Tachi, and L. Mahadevan. 2016. Programming curvature using origami tessellations. *Nature Materials* 15, 5 (2016), 583–588.

Gerald Farin. 2002. *Curves and surfaces for CAGD: a practical guide.* Morgan Kaufmann.

Akash Garg, Andrew O Sageman-Furnas, Bailin Deng, Yonghao Yue, Eitan Grinspun, Mark Pauly, and Max Wardetzky. 2014. Wire mesh design. *ACM Transactions on Graphics* 33, 4 (2014).

Ruslan Guseinov, Connor McMahan, Jesús Pérez, Chiara Daraio, and Bernd Bickel. 2020. Programming temporal morphing of self-actuated shells. *Nature communications* 11, 1 (2020), 1–7.

Ruslan Guseinov, Eder Miguel, and Bernd Bickel. 2017. CurveUps: Shaping Objects from Flat Plates with Tension-Actuated Curvature. *ACM Trans. Graph.* 36, 4, Article 64 (7 2017), 12 pages. https://doi.org/10.1145/3072959.3073709

Christian Hafner and Bernd Bickel. 2021. The design space of plane elastic curves. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–20.

Florin Isvoranu, Julian Panetta, Tian Chen, Etienne Bouleau, and Mark Pauly. 2019. X-Shell Pavilion: A Deployable Elatic Rod Structure. *Proceedings of the IASS Annual Symposium - Structural Membranes 2019* (2019).

David Jourdan, Melina Skouras, Etienne Vouga, and Adrien Bousseau. 2022. Computational Design of Self-Actuated Surfaces by Printing Plastic Ribbons on Stretched Fabric. *Computer Graphics Forum* 41, 2 (April 2022). http://www-sop.inria.fr/reves/Basilic/2022/JSVB22

Mina Konaković-Luković, Julian Panetta, Keenan Crane, and Mark Pauly. 2018. Rapid deployment of curved surfaces via programmable auxetics. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–13.

Francesco Laccone, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Tim Schork. 2021. Integrated computational framework for the design and fabrication of bending-active structures made from flat sheet material. *Structures* 34 (dec 2021), 979–994. http://vcg.isti.cnr.it/Publications/2021/LMPCS21

Riccardo LaMagna and Jan Knippers. 2017. On the behaviour of bending-active plate structures. *Proceedings of the IASS Annual Symposium 2017 "Interfaces: architecture, engineering, science* (2017).

J. Lienhard. 2014. *Bending-Active Structures: Form-finding Strategies Using Elastic Deformation in Static and Kinetic Systems and the Structural Potentials Therein.* Universität Stuttgart Inst. f. Tragkonstr.

Julian Lienhard and Christoph Gengnagel. 2018. Recent developments in bending-active structures. In *Proceedings of IASS Annual Symposia*, Vol. 2018. International Association for Shell and Spatial Structures (IASS), 1–8.

Bosheng Liu and Tanvir R. Faisal. 2022. Computational Design and Fabrication of a Bending-Active Structure Using Fiberglass: A Bioinspired Pavilion Mimicking Marine Microorganism Radiolaria. *Journal of Bionic Engineering* 19 (2022).

Luigi Malomo, Jesús Pérez, Emmanuel Iarussi, Nico Pietroni, Eder Miguel, Paolo Cignoni, and Bernd Bickel. 2018. FlexMaps: Computational Design of Flat Flexible Shells for Shaping 3D Objects. *ACM Trans. on Graphics - Siggraph Asia 2018* 37, 6 (nov 2018), 14. http://vcg.isti.cnr.it/Publications/2018/MPIPMCB18 https://doi.org/10.1145/3272127.3275076.

Vittorio Megaro, Jonas Zehnder, Moritz Bächer, Stelian Coros, Markus H Gross, and Bernhard Thomaszewski. 2017. A computational design tool for compliant mechanisms. *ACM Trans. Graph.* 36, 4 (2017), 82–1.

Saurabh Mhatre, Elisa Boatti, David Melancon, Ahmad Zareei, Maxime Dupont, Martin Bechthold, and Katia Bertoldi. 2021. Deployable Structures Based on Buckling of Curved Beams Upon a Rotational Input (Adv. Funct. Mater. 35/2021). *Advanced Functional Materials* 31, 35 (2021), 2170261. https://doi.org/10.1002/adfm.202170261 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/adfm.202170261

Jorge Nocedal and Stephen Wright. 2006. *Numerical optimization*. Springer Science & Business Media.

Julian Panetta, Florin Isvoranu, Tian Chen, Emmanuel Siéfert, Benoît Roman, and Mark Pauly. 2021. Computational inverse design of surface-based inflatables. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–14.

Julian Panetta, MINA Konaković-Luković, Florin Isvoranu, Etienne Bouleau, and Mark Pauly. 2019. X-shells: A new class of deployable beam structures. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–15.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019), 8026–8037.

Jesús Pérez, Miguel A Otaduy, and Bernhard Thomaszewski. 2017. Computational design and automated fabrication of kirchhoff-plateau surfaces. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–12.

Stefan Pillwein, Kurt Leimer, Michael Birsak, and Przemyslaw Musialski. 2020. On elastic geodesic grids and their planar to spatial deployment. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 125–1.

Stefan Pillwein and Przemyslaw Musialski. 2021. Generalized Deployable Elastic Geodesic Grids. *ACM Trans. Graph.* 40, 6, Article 271 (dec 2021), 15 pages. https://doi.org/10.1145/3478513.3480516

G. Quinn and Christoph Gengnagel. 2014. A review of elastic grid shells, their erection methods and the potential use of pneumatic formwork. In *Mobile and rapidly assembled structures*, Vol. 136.

Yingying Ren, Uday Kusupati, Julian Panetta, Florin Isvoranu, Davide Pellis, Tian Chen, and Mark Pauly. 2022. Umbrella Meshes: Elastic Mechanisms for Freeform Shape Deployment. *ACM Trans. Graph.* 41, 4 (2022).

Yingying Ren, Julian Panetta, Tian Chen, Florin Isvoranu, Samuel Poincloux, Christopher Brandt, Alison Martin, and Mark Pauly. 2021. 3D weaving with curved ribbons. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–15.

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

Jonas Schikore and Eike Schling. 2021. The Kinetic Umbrella: Designing a transformable semi-elastic grid structure. In *10th edition of the conference on Textile Composites and Inflatable Structures (STRUCTURAL MEMBRANES 2021)*.

Jonas Schikore, Eike Schling, Thomas Oberbichler, and Anna Bauer. 2021. Kinetics and Design of Semi-Compliant Grid Mechanisms. In *Advances in Architectural Geometry (AAG)*.

Eike Schling, Martin Kilian, Wang Hui, Jonas Schikore, and Helmut Pottmann. 2018. Design and Construction of Curved Support Structures with Repetitive Parameters. In *Advances in Architectural Geometry (AAG)*.

E. Soriano, R. Sastre, and D. Boixader. 2019. G-shells:flat collapsible geodesic mechanisms for gridshells. In *International Conference Form and Force / Symposium of the International Association for Shell and Spatial structures / International Conference on Textile Composites and Inflatable Structures*. International Centre for Numerical Methods in Engineering (CIMNE), 1894. http://hdl.handle.net/2117/330614

Josh Vekhter, Jiacheng Zhuo, Luisa F Gil Fandino, Qixing Huang, and Etienne Vouga. 2019. Weaving Geodesic Foliations. *ACM Trans. Graph.* 38, 4, Article 34 (jul 2019), 22 pages. https://doi.org/10.1145/3306346.3323043

Richard A Waltz and Jorge Nocedal. 2004. KNITRO 2.0 User's Manual. *Ziena Optimization, Inc.[en ligne] disponible sur http://www. ziena. com (September, 2010)* 7 (2004), 33–34.