EPFL

Thèse n° 10 342

Computational Design of Deployable Gridshells with Curved Elastic Beams

Présentée le 31 janvier 2025

Faculté informatique et communications
Laboratoire d'informatique géométrique
Programme doctoral en informatique et communications

pour l'obtention du grade de Docteur ès Sciences

par

Quentin Christian BECKER

Acceptée sur proposition du jury

Prof. N. Boumal, président du jury
Prof. M. Pauly, directeur de thèse
Prof. D. Panozzo, rapporteur
Prof. E. Schling, rapporteur
Dr B. Thomaszewski, rapporteur

École polytechnique fédérale de Lausanne

2025

Quentin Becker
ORCID: 0000-0003-0952-7063

Acknowledgements

I would first like to express my profound gratitude to my advisor, Mark Pauly, for his guidance, support, and encouragement throughout my Ph.D. journey. Thanks for introducing me to the field of computational design and accompanying me on my initial research steps. I am also grateful for the freedom you gave me to explore my ideas and for the trust you placed in me. This experience would not have been so much fun if it were not for the countless discussions we had on inflatable-auxetic-C-shells or perhaps a permutation of those words. I am also thankful for the opportunity to collaborate with others and meet wonderful people. Thanks for always striving for excellence and beauty, and transmitting this aspiration to me.

I would like to extend a special thank you to Seiichi Suzuki, without whom most of the creations presented in this thesis would not have existed. Thank you for your unfailing patience, enthusiasm, and kindness. I am grateful for the invaluable time spent together trying to understand our capricious gridshells, and during trips to conferences. Thanks for being a friend and a great collaborator.

My research would not have been feasible without the support and the incredible work of my collaborators: Seiichi Suzuki, Uday Kusupati, Samara Ren, Davide Pellis, Julian Panetta, Filip Goč, Florin Isvoranu, and Mike Yan Michelis.

I would like to express my gratitude Urs Bergmann for hosting me exceptionally well during my internship at Google Research. Thanks for the great care you took in supervizing me, and thanks to the rest of the team for the fruitful exchanges we had together.

I am thankful to the rest of my thesis committee members, Nicolas Boumal, Daniele Panozzo, Eike Schling, and Bernhard Thomaszewski, for their contributions in improving this thesis and for the insightful discussions we had during the private defense.

My time at EPFL in the Geometric Computing Lab (GCM) could not have been more enjoyable thanks to the wonderful lab members I had the chance to work with or discuss with: Samara Ren, Uday Kusupati, Michele Vidulis, Ziqi Wang, Desmond Liu, Liliane-Joy Dandy, Seiichi Suzuki, Julian Panetta, Oliver Gross, Klara Mundilova, Christopher Brandt, Tim Chen, Ulysse Martel, and Robin Jodon. Thank you to all of you. I am particularly grateful for the unwavering support Madeleine Robert provided, the discussions about travel experiences, skiing, and life in general. I had the chance to start my Ph.D. with Samara Ren, and I am grateful for the endless discussions we had about more or less technical aspects of our work, especially when it came to purring creatures.

Uday Kusupati, Michele Vidulis, and Martina Massardi were always up for our crazy biking trips, and I want to thank them for making those adventures possible. I would also like to thank our lab neighbors from RGL for bringing life to our floor and sharing pizza parties among other activities.

I would like to thank my friends in Switzerland, expecially Sylvain, Arnout, Etienne, Les Frouzes, and also those who have been bearing with me since before my Ph.D. journey began: the Team Rocket, the Lausanne's Champion's Project's, and everyone else. Thanks for the wonderful times and laughter we shared; you made my Ph.D. experience truly unforgettable.

Je garde le mot de la fin pour ma famille, mes frères Thibault et Yann et mes parents, pour leur soutien indéfectible, les moments inoubliables à marcher, courir, plancher ou juste rire, et surtout pour leur amour inconditionnel. Merci pour tout.

Q.B.

Abstract

Deployable gridshells are lightweight structures made of interconnected elastic beams. They can be actuated from a compact state to a freeform and volume-enclosing deployed shape. This thesis introduces *C-shells*, a novel class of deployable gridshells, which employs curved elastic rods connected at single-axis rotational joints. As opposed to their straight counterparts, C-shells are guaranteed to be assembled in a planar and stress-free configuration while showing a wide diversity in their deployed shapes. They may serve as temporary shelters, pavilions, or on a smaller scale, as deployable furniture or decorative elements.

This thesis presents a comprehensive framework for the forward exploration of C-shell designs, enabling designers to interactively search the shape space and generate deployable structures with diverse appearances and topologies. The framework combines human-interpretable manipulations of a reference linkage with an efficient physics-based simulation to predict the deployed shape and mechanical behavior of the structure. Preservation of the linkage deployability and smoothness of the edits are ensured through the use of conformal maps as design handles. The framework is implemented as a Rhino-Grasshopper plugin, providing visual and quantitative real-time feedback on the deployed state.

The inverse design of C-shells is also addressed, where the deployed shape is given, and the flat state of the structure is computed. This thesis introduces a two-step pipeline composed of a flattening method and a design optimization algorithm. The flattening algorithm is based on kinetic considerations underlying the deployment of C-shells. The method harmonizes a flat and a hypothetical deployed state constrained on a user-prescribed target surface. The flat beam layout is further adjusted to minimize the deviation of the deployed shape to the target surface while ensuring a low elastic energy deployed state, under some beam smoothness regularization. The proposed method is validated through scanned small-scale prototypes.

C-shells are made of curved rods, which entails additional material waste compared to straight beams. To address this issue, this thesis presents a rationalization method that splits the curved beams into smaller straight elements which can be grouped into a sparse *kit of parts*, while preserving user-provided designs. The original combinatorial problem of jointly assigning parts to elements and adapting the parts' geometry is relaxed into a two-step optimization process incorporating our physics-based simulation, making it tractable using continuous optimization techniques. The proposed method applies more generally to bending-active structures and is further

demonstrated on orthogonal gridshells and umbrella meshes. Part reuse is assessed in a study of the trade-off between the number of parts and fidelity to the input designs.

Keywords: deployable gridshells, physics-based simulation, numerical optimization, geometric modeling, computational inverse design, fabrication-driven design.

Résumé

Les gridshells déployables sont des structures légères composées de poutres élastiques interconnectées. Elles peuvent être actionnées pour passer d'un état compact à une forme déployée libre et enveloppant un volume. Cette thèse présente les *C-shells*, une nouvelle classe de gridshells déployables, qui utilise des tiges élastiques incurvées reliées à des joints rotatifs à axe unique. Contrairement à leurs homologues composés de tiges droites, les C-shells sont garanties d'être assemblées dans une configuration plane et sans contrainte, tout en montrant une grande diversité dans leurs formes déployées. Elles peuvent servir d'abris temporaires, de pavillons ou, à plus petite échelle, de meubles déployables ou d'éléments décoratifs.

Cette thèse présente un cadre complet pour l'exploration de designs de C-shells, permettant aux concepteurs de sonder l'espace de formes de manière interactive et de générer des structures déployables d'apparences et de topologies diverses. Le cadre combine des manipulations interprétables par l'homme d'un assemblage de référence avec une simulation physique efficace pour prédire la forme déployée et le comportement mécanique de la structure. La préservation de la capacité de déploiement de l'assemblage et la régularité des modifications sont assurées par l'utilisation de fonctions conformes comme leviers de conception. Les outils développés sont rassemblés en un plugin Rhino-Grasshopper, fournissant un retour visuel et quantitatif en temps réel de l'état déployé de la structure.

La conception inverse des C-shells est également abordée, où la forme déployée est donnée et l'état aplati de la structure est calculé. Cette thèse introduit un processus en deux étapes composé d'une méthode d'aplatissement et d'un algorithme d'optimisation de design. L'algorithme d'aplatissement est basé sur des considérations cinématiques sous-jacentes au déploiement des C-shells. La méthode harmonise un état plat et un état déployé hypothétique contraint à une surface cible prescrite par l'utilisateur. La disposition des tiges dans le plan est ensuite ajustée pour minimiser la déviation de la forme déployée à la surface cible tout en garantissant un état déployé à faible énergie élastique, en régularisant également la courbure au repos des tiges. La méthode proposée est validée par des prototypes scannés de petites dimensions.

Les C-shells sont constituées d'éléments incurvés, ce qui induit des résidus de matériaux supplémentaires par rapport aux poutres droites. Pour résoudre ce problème, cette thèse présente une méthode de rationalisation qui divise les poutres courbes en éléments droits plus petits qui peuvent être regroupés en un *jeu de pièces* de taille restreinte, tout en préservant les designs fournis

par l'utilisateur. Le problème combinatoire initial de l'affectation conjointe des pièces aux éléments et de la modification de la géométrie des pièces est relaxé en un processus d'optimisation en deux étapes incorporant notre simulation physique, ce qui le rend traitable à l'aide de techniques d'optimisation continue. La méthode proposée s'applique plus généralement aux structures actives en flexion et fait l'objet d'une démonstration supplémentaire sur des gridshells orthogonales et des maillages parapluie. La réutilisation des pièces est évaluée dans une étude du compromis entre le nombre de pièces et la fidélité aux designs d'entrée.

Mots-clés : gridshells déployables, simulation physique, optimisation numérique, modélisation géométrique, conception inverse computationnelle, conception guidée par la fabrication.

Published Content

This thesis is based on the following publications:

- **Quentin Becker**, Seiichi Suzuki, Yingying Ren, Davide Pellis, Julian Panetta, Mark Pauly. C-shells: Deployable Gridshells with Curved Beams. *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia 2023)*, Best Paper Award Honorable Mention.
- **Quentin Becker**, Seiichi Suzuki, Mark Pauly. Interactive Design of C-shells Using Reduced Parametric Families. *International Association for Shell and Spatial Structures (IASS) (Proc. of the IASS Annual Symposium 2024)*.
- **Quentin Becker***, Uday Kusupati*, Seiichi Suzuki, Mark Pauly (*joint first authors). Computational Design of a Kit of Parts for Bending-Active Structures. *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia 2024)*.

Additionally, the author contributed to the following publication during the same period:

- Mike Yan Michelis, **Quentin Becker**. On Linear Interpolation in the Latent Space of Deep Generative Models. *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*, Spotlight.

The abstract of the additional paper is included in Appendix C.

Contents

List of Illustrations

List of Tables

Chapter 1

Introduction

> A work of architecture may be both dramatic and organic, but it will fail to achieve the perfection of a work of art unless it be also *schematic*. It (this word) means a disposition of parts according to some co-ordinating law or principle.
>
> *Claude Bragdon, 1931*

Architecture has continuously evolved through the ages, taking different forms which were often dictated by the contingencies of the time and the available materials. From vernacular structures to the most sophisticated contemporary buildings, architects have designed habitats, bridges, monuments, gathering spaces, and edifices of prime importance that shape our environment and our lives.

First, architecture must answer a need in the most efficient manner. Jean-Nicolas-Louis Durand was one of the first to propose a rational and quantitative approach to architecture in 1809, advocating for two core principles: Convenience and economy. While convenience refers to the functional aspect of a structure, economy can be understood as the efficient use of resources or the simplicity of the design. This idea has been further developed by the modernist movement, spearheaded by Louis Sullivan who famously coined the credo "form ever follows function" which establishes a clear hierarchy of concepts.

Expressionist, Art Deco, then post-modernist movements emerged from the perceived soberness and uniformity of modern architecture. Through different means of expression, both sought to reintroduce ornamentation, symbolism, and historical references in architecture. Robert Venturi stated in his seminal book *Complexity and Contradiction in Architecture* 1977 that "When circumstances defy order, order should bend or break: Anomalies and uncertainties give validity to architecture.", advocating for a more eclectic approach to design.

More recently, the growing awareness of the current environmental crisis caused ecological considerations to percolate into the architectural discourse. This led to the development of sustainable design practices and the creation of new movements such as ecomodernism or architecture

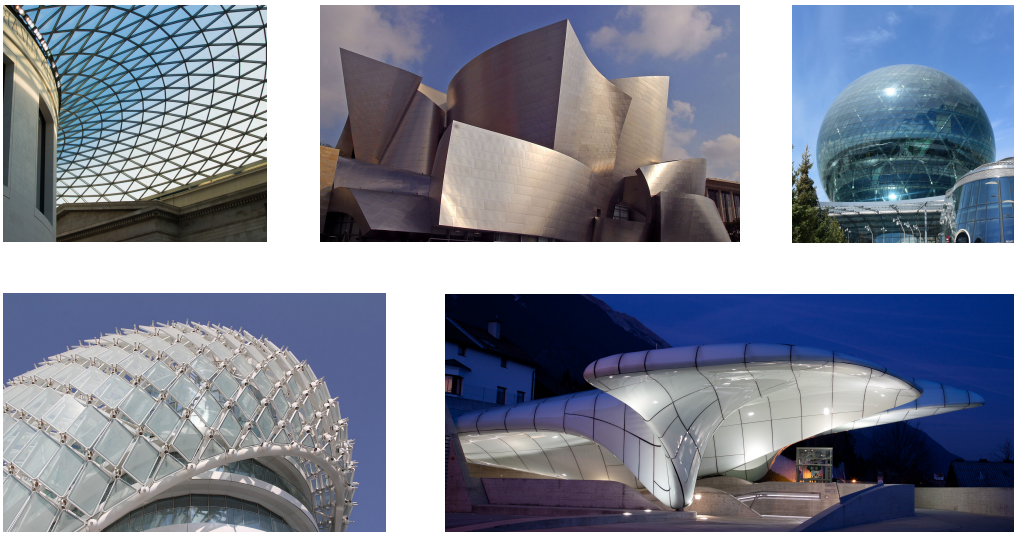


Figure 1.1: Examples of freeform surface paneling in contemporary architecture. From top to bottom and left to right: The Great Court Roof of the British Museum in London, the Walt Disney Concert Hall in Los Angeles, the Nur Alem Museum of Future Energy in Astana, the Yas Island Hotel in Abu Dhabi, and the Hungerburg funicular railway in Innsbruck. Photo credits can be found in Appendix C.

of reuse. The latter posits deconstruction and durability as the two main guiding principles in architecture [Gorgolewski, 2017].

Architecture is a fluctuating domain, and is ever reinvented through the social and technological changes of the time. However, the tension between function and aesthetics remains. Because of its scientific nature, this thesis seems to push in the direction of rationalization of architecture. However, the intent is merely to provide the quantitative tools necessary for an architect to express their creativity on a challenging and capricious class of structures: Deployable gridshells.

Design Through Form: Architectural Geometry

Geometry has always occupied a central place in architecture. As a first example Pythagorean triples can be found in the proportions of ancients temples such as the Parthenon [Chiotis, 2021]. The catenary curve can be retrieved across different architectural styles, from the Gothic vaults of the King's College Chapel in Cambridge, to the Baroque dome of Saint Paul's Cathedral in London, or the Gateway Arch in St. Louis (technically a weighted catenary).

Contemporary architecture pushes the notion of harmony and symmetry to its limits with, for

instance, the Great Court Roof of the British Museum in London from Buro Happold and Chris Williams as primary engineer, or the Hungerbur funicular railway designed by Zaha Hadid Architects, see Figure 1.1. Approximating freeform surfaces is a central and common challenge to pieces of work affiliated with this movement. The physical realization of such complex shapes requires meeting structural constraints that are often of geometric nature. As a consequence, a new domain of architectural research emerged following the pioneering work of Pottmann and co-workers [Pottmann et al., 2007a] and took the name of *Architectural Geometry*.

Among other works, Frank Gehry's Walt Disney Concert Hall in Los Angeles takes the form of a freeform developable surface, which allows rolling out sheet metal bands to cover the shape [Shelden, 2002]. The concept of parallelism for polyhedral surfaces is central to the design of support structures with torsion-free nodes [Pottmann et al., 2007b]. The outer shell of the Yas Island Hotel in Abu Dhabi applies this geometric concept to ensure torsion-free nodes.

Panelization of freeform surfaces consists in approximating a continuous surface with a set of panels. Constraints on the panel shapes motivated by economical or aesthetic reasons led to interesting variations of the original problem. Imposing planarity on the elements restricts surfaces to quad meshes with planar faces (PQ meshes), and led to the notion of Conical Meshes [Liu et al., 2006]. On the other hand of the spectrum, the manufacturing technique of cold bending of flat panels allows for the fabrication of doubly curved elements. Such panels are optimized for the realized façade to appear smooth while generating minimal physical stress on its elements [Gavriil et al., 2020]. In an attempt to slash the number of molds needed for fabrication Pellis and colleagues 2021 introduce a shape projection algorithm that turns a freeform surface into a Weingarten surface. Spherical panels recently led to the notion of meshes with spherical faces [Kilian et al., 2023, Cisneros et al., 2024], and have been demonstrated with the Nur Alem Museum of Future Energy in Astana.

Complexity Emerging from Simplicity: Deployable Elastic Gridshells

Buckling, a physical instability common to slender structures under compression, is usually proscribed in architecture. It often shows as a failure of a structure. However, when properly harnessed, buckling can be used to achieve complex shapes with minimal material requirements. The Thematic Pavilion Expo in Yeosu designed by Soma Architects and engineered by Knippers Helbig Advanced Engineering is a full-scale example of a kinetic façade [Knippers et al., 2012]. Motion is enabled by the flexibility of its elements and the reversibility of their elastic deformations.

Figure 1.2: Examples of three large-scale gridshells. From left to right: The Multihalle in Mannheim, the Weald and Downland Gridshell Building in Singleton, and the Savill Building in Windsor. Photo credits can be found in Appendix C.

Bending-Active Structures

Structures employing buckling and bending of its elements to ensure the static equilibrium of the whole are coined *bending-active structures* [Knippers et al., 2011]. As opposed to their rigid counterparts, of which the paneling applications mentioned above are a subset, bending-active structures tend to escape simple geometric descriptions. Physical prototyping in the form of hanging chain models or soap films were then a common practice to explore the equilibrium shapes of such structures, as performed by Frei Otto [Otto and Rasch, 1995] or Heinz Isler [Boller et al., 2024]. These experimental approaches were complemented by numerical *form-finding* methods, which allowed for a more efficient exploration of shapes [Lienhard et al., 2013]. Initiated in the 60s, these efforts led to the notable force density method [Schek, 1974] and dynamic relaxation method [Barnes, 1977]. Despite the limited computational resources of the time, such techniques contributed to the numerical load certification of the pioneering Multihalle in Mannheim, whose roof was designed by Frei Otto and Ove Arup & Partners. Notably, early small-scale physical prototypes allowed detecting insufficient load-bearing capabilities and led to doubling the grid layer [Happold and WI, 1975], demonstrating the synergy between physical and numerical form-finding methods.

Elastic Gridshells

Elastic gridshells, similar to the Multihalle, are structures composed of elastic beams connected at their crossings through joints of various kinds. Their erected state exploits the combined deformation of the rods due to constraints imposed at the boundary. They are bending-active. In addition to being lightweight with conveniently transportable structural components, its timber elements are shaped then deformed, greatly simplifying the fabrication process. In contrast, steel and concrete shells must be formed according to the final design. Aside the Multihalle, the Weald and Downland Gridshell Building in Singleton by Buro Happold and Edward Cullinan Architects

[Harris et al., 2003], and the Savill Building in Windsor by Buro Happold, Engineers Haskins Robinson Waters, and Glen Howells Architects [Harris and Roynon, 2008], are two of the most remarkable large-scale examples, see Figure 1.2.

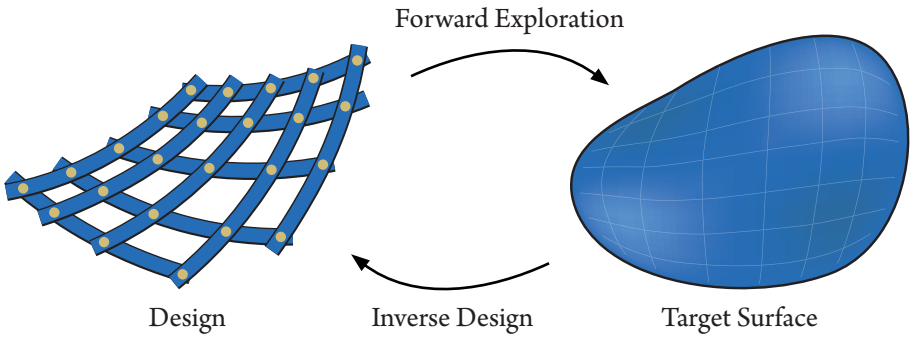


Figure 1.3: Forward simulation aims at finding the equilibrium shape of a structure given its geometry and material properties. The inverse problem updates the geometry of the structure given a target shape. In this thesis, the material parameters are fixed during inverse design.

Baek and coworkers 2018 introduced a physics-based forward exploration tool (see Figure 1.3) supplemented with a geometric rationalization of the design space of elastic gridshells. This geometric analysis is based on the assumption of rods inextensibility, which allows regarding gridshells as Chebyshev nets. Elastic gridshells typically assume a *regular layout* of beams and an *extrinsic* erection process. In this context, regularity is meant in terms of spacing between beams crossing: The structure has undeformed elements in its assembly state. Extrinsic erection means that the structure's final state is determined by the external forces applied to it, typically through boundary constraints. Once erected, the lattice of beams transmits the loads along the laths directions through bending, stiffening the structure along these axes. However, such structures cannot resist loads along the diagonals of the lattice, as opposed to continuous shells. Diagonal bracing and making joint rigid are common strategies to overcome this limitation.

Deployable Elastic Gridshells

The introduction of X-shells by Panetta and colleagues 2019 brought a new perspective on gridshells, allowing them to deploy from a compact state to a 3D state. They waive both regularity constraints in the grid layout and the need for extrinsic erection: The deployed shape is directly encoded in the irregular assembly state. Additionally, the assembly and deployment phases of X-shells are decoupled, where gridshells require the elements to be connected incrementally in the erected state. This simplifies the construction process. Automatic inverse-design tools are

developed to optimize linkages so they can be assembled with minimum stress and achieve a target shape in their deployed state. However, the full inverse problem illustrated in Figure 1.3 is not solved and lacks an efficient design initialization strategy. Deployable Elastic Geodesic Grids developed by Pillwein and coworkers observe that ribbon-like beams trace geodesics on a surface and propose a geometrically-driven inverse design to find the best layout of beams on a given surface [Pillwein et al., 2020, 2021, Pillwein and Musialski, 2021]. For the net to flatten in a stress-free manner, they introduce notches along the beams that allow for sliding of the joints. This thesis introduces C-shells, deployable gridshells with curved elastic beams, an extension of X-shells. These new degrees of freedom allow deviating from geodesics at no cost on the elastic energy in the deployed state by introducing geodesic curvature in the rest shape of the beams. While our method allows for a wide range of cross-sections, we focus on beam profiles with a clear stiffer bending direction e.g., thin rectangular cross-sections, which we align tangent to the surface the structure approximates. More details will be provided in the following chapters.

Alternatively, Schling and colleagues 2018 propose to orient the strong axis of the beams orthogonal to the target surface and allow for a higher normal load-bearing capability. This approach is based on asymptotic networks and the resulting class of structures is called *A-shells*. The shape space is restricted to surfaces with negative Gaussian curvature, for which two distinct families of lines of constant normal curvature i.e., the *asymptotic lines*, exist. Imposing the beams to meet at a constant right node angle requires surfaces to have Constant Mean Curvature (CMC). Straight beams additionally impose the surface to be minimal i.e., its mean curvature vanishes. Liu and coworkers 2023a further study the properties the target surface must benefit e.g., linear Weingarten surface of hyperbolic type for quadrilateral meshes, for the corresponding gridshell to collapse. A-shells and their variations further illustrates the tight connection between geometry and structural behavior in bending-active structures.

Other Deployable Systems

Bending-active structures are typically composed of simple atomic elements which, when combined in the right way, give rise to complex and expressive shapes. This principle can be found across different physical systems in the computational fabrication and computer graphics literature. Umbrella meshes [Ren et al., 2022], auxetic materials [Konaković et al., 2016], and bistable auxetic surface structures [Chen et al., 2021] are some examples of shape morphing systems relying on conformal flattening to extract the shape of its constituent elements from a target surface. While each cell of the structure behaves in an elementary way, incompatibilities introduced by combining them lead to the emergence of complex global behaviors.

Towards Sustainability: Kit-of-Parts Design

In an effort to reduce the environmental impact of construction repetition, sparsity, and reuse of elements are key. Following these principles prohibits the use of custom-made elements, which can hardly be transferred from one structure to another, and promotes simplicity. Harmony and symmetry, concepts that are dear to architects, are now transferred from the shape itself to the constitutive elements of the structure. As mentioned above, simplicity is at the core of bending-active structures principles and provides a fertile ground for their rationalization into a sparse set of parts i.e., a *kit-of-parts*.

Taking the simplest elements at hands, sticks connected with rigid joints, Brütting and colleagues demonstrate that it is possible to preserve diversity in the achievable assemblies while maintaining a very limited set of parts [Brütting et al., 2021]. At the other end of the spectrum, Reconfigurable Umbrella Meshes (RUM) [Kusupati et al., 2023] introduce a unique fully reconfigurable part. Such assemblies have the notable property that their assembly state is stress-free no matter the elements' configuration, and allow for a wide range of achievable shapes using a unique part. However, this comes at the cost of more complex elements. A crucial trade-off inherent to the design of a kit of parts is then the balance between parts complexity and the size of the kit. Our thesis explores this trade-off in the context of the larger class of bending-active structures.

As a response to Louis Sullivan's quote, Brütting and colleagues suggest that "form follows availability" [Brütting et al., 2019], effectively shifting the focus from the shape of the structure to the elements that compose it. Availability from existing stock of parts are explicitly determined at the start of the design process, which qualifies their work as a *pre-rationalization* approach. Austern and co-workers 2018 define pre-rationalization as the process of guiding the design process based on known fabrication constraints. We propose in Chapter 5 a *co-rationalization* method where some fabrication constraints serve as handles for the user to guide the design process.

The ideas of reuse and repetitions have also emerged in other domains of architecture. Paneling freeform surfaces may incorporate notions of mold reuse of as exposed by Eigensatz and colleagues [Eigensatz et al., 2010b]. Panels are grouped according to geometric characteristics and further clustered within each category. Interlocking assemblies of rigid blocks may also benefit from encouraging repetition in the structure as shown in the context of masonry shells [Chen et al., 2023] for instance. Sculptures resembling the Art Tower in Mito by Arata Isozaki can be obtained from a sparse set of triangles through shape optimization as presented by Liu and coworkers 2021.

Using natural materials is another way to reduce the environmental impact of construction. Bamboo, for instance, has elastic material properties that are well-suited for bending-active structures. Suzuki and co-workers 2023 designed the first large-scale prototype of a unique family of bending-active bamboo structures made of deployable cylinders connected and woven at predefined nodes.

1.1 Contributions

A summary of the main contributions of this thesis is listed below.

- We introduce a novel class of deployable gridshell structures, *C-shells*, composed of curved elastic beams connected at single-axis rotational joints. C-shells can be assembled in a stress-free planar configuration and deploys into a 3D shape when actuated.
- We propose a unified treatment of the design optimization of gridshell-like structures, and demonstrate its efficiency on three kinds of bending-active structures: C-shells, C-shells discretized into a kit of parts, and orthogonal grids discretized into a kit of parts.
- We provide a new flattening algorithm as a component of the inverse design pipeline of C-shells and demonstrate its effectiveness on a variety of target surfaces.
- We present families of curve layouts based on conformal maps that allow for efficient and interactive design exploration of C-shells with varying topologies.
- We introduce a smooth relaxation of the combinatorial task of rationalizing a set bending-active structures into a sparse *kit of parts*, and apply it on C-shells, Orthogonal Grids, and Umbrella Meshes.
- We quantitatively demonstrate the accuracy of our simulation framework on fabricated prototypes. While the computational tools developed are primarily intended for a use in architecture, this thesis does not aim to solve the engineering challenges related to scaling the structures up to a building scale.

1.2 Outline

The thesis is organized as follows:

- In Chapter 2, we build the necessary tools used in the rest of the thesis in a bottom-up fashion. We start by introducing beams as discretized curves deforming elastically. Equilibrium is cast as a variational problem solved through minimization of the total elastic energy under constraints. We then introduce the concept of deployable gridshells and elaborate on their actuation and inverse design.
- In Chapter 3, we introduce C-shells, a new class of deployable gridshells with curved elastic beams. We present a computational design framework for C-shells that integrates accurate forward simulation in the inner loop of a design optimization framework. The optimization algorithm either takes a user-designed planar layout through forward exploration, or the output of a planarization algorithm we develop for the inverse design problem.
- In Chapter 4, we list kinetic properties of C-shells and study the sensitivity of the deployed state to the beam layout. We subsequently construct intuitive forward exploration tools, mainly based on conformal maps, for the user to interactively design C-shells.
- In Chapter 5, the problem of rationalizing bending-active structures into a kit of parts is formalized and addressed. We propose a relaxation of the otherwise combinatorial problem into a two-stage continuous optimization problem that can be solved efficiently. We develop part-reuse metrics and study the trade-off between the number of parts and the preservation of input designs.
- In Chapter 6, we conclude the thesis and discuss future research directions.

Chapter 2

Background

This chapter provides insights into the mathematical and physical principles that underlie the design of deployable elastic gridshells. Based on a geometric description of the quantification of deformations of elastic curves, we cast the problem of finding equilibrium states of elastic curves under external forces as an optimization problem. Assemblies of elastic beams in the form of gridshells are then introduced, and the energy of such systems is defined. Our gridshell structures are deployable and their erected state are defined as minimizers of their total elastic energy under external constraints. Due to their unintuitive and complex behavior, deployable gridshells are hardly steerable through manual design of their constitutive elements. We devise a strategy based on the sensitivity of a prescribed quality metric with respect to the design variables to optimize the geometry of the gridshell elements.

2.1 Elastic Curves

Curved elastic beams are the building blocks of our deployable gridshells. We briefly introduce the Discrete Elastic Rods model [Bergou et al., 2010] to describe the deformation of elastic beams.

Geometry of Continuous Curves

A curve is a continuous map $\gamma : [0, 1] \to \mathbb{R}^3$. We assume that γ is sufficiently *smooth*, i.e., it has continuous first and second derivatives. The arc-length $t \mapsto l(t)$ of the curve is defined as

$$l(t) = \int_0^t \left\| \frac{\mathrm{d}\gamma}{\mathrm{d}u} \right\| \mathrm{d}u,$$

where $\|\cdot\|$ denotes the Euclidean norm, and $\frac{\mathrm{d}\gamma}{\mathrm{d}u}$ is a vector tangent to the curve at position $\gamma(u)$ for some $u \in [0, 1]$. The arc-length parameter s is defined as $s = l(t)$ for some parameter $t \in [0, 1]$. Assuming the curve to be *regular*, i.e., it has non-vanishing derivatives everywhere, we can re-parameterize the curve by its arc-length into a curve $s \mapsto \tilde{\gamma}(s) = \gamma\left(l^{-1}(s)\right)$ for any $s \in [0, L]$, where $L = l(1)$ is the length of the curve. The length function l is indeed strictly increasing and continuous and thus invertible. The tangent vector $\mathbf{t}(s)$ is defined as

$$\mathbf{t}(s) = \frac{\mathrm{d}\tilde{\gamma}}{\mathrm{d}s}.$$

As a consequence of the re-parameterization, the tangent vector is a unit vector, i.e., $\|\mathbf{t}(s)\| = 1$. The curvature $\kappa(s)$ measures the rate of change of the tangent vector along the curve and is defined as

$$\kappa(s) = \left\| \frac{\mathrm{d}\mathbf{t}}{\mathrm{d}s} \right\|.$$

The normal vector gives the direction of tangent vector variation as

$$\mathbf{n}(s) \propto \frac{\mathrm{d}\mathbf{t}}{\mathrm{d}s},$$

where the proportionality constant is chosen such that $\mathbf{n}(s)$ is a unit vector excluding inflection points i.e., points of vanishing curvature, where $\mathbf{n}$ is undefined.

The triplet defined by the tangent vector $\mathbf{t}(s)$, the normal vector $\mathbf{n}(s)$, and the binormal vector $\mathbf{b}(s) = \mathbf{t}(s) \times \mathbf{n}(s)$ is called the *Frenet frame* of the curve. By construction, the frame is *adapted* to the curve: One of the vectors of the frame is tangent to the curve all along the curve.

However, the Frenet frame is not uniquely defined at inflection points where the curvature κ vanishes. This motivates the use of a different frame, which was developed by [Bishop, 1975]. The orthogonal Bishop frame $(\underline{\mathbf{d}}_1(s), \underline{\mathbf{d}}_2(s), \underline{\mathbf{d}}_3(s))$ shown in Figure 2.1 is adapted i.e., $\underline{\mathbf{d}}_1(s) = \mathbf{t}(s)$ for $s \in [0, L]$, and is defined such that it is twist-free.

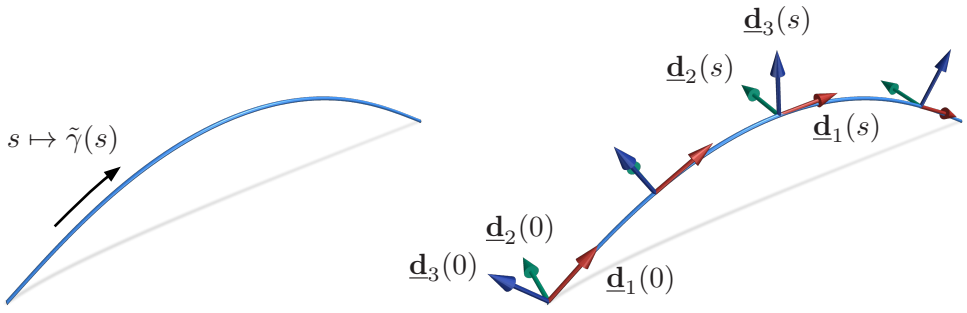


Figure 2.1: A smooth arc-length parameterized curve $\tilde{\gamma}$ (left) and its associated Bishop (or reference) frame (right).

The twist of the Frenet frame $\tau(s)$, which is also the torsion of the curve, is defined as

$$\tau(s) = \frac{\mathrm{d}\mathbf{n}}{\mathrm{d}s} \cdot \mathbf{b} = -\mathbf{n} \cdot \frac{\mathrm{d}\mathbf{b}}{\mathrm{d}s}. \tag{2.1}$$

Whenever the twist of the Frenet frame vanishes for a curve free of inflection points, the normal vector and consequently the tangent vector are only allowed to rotate in the plane they form

along the curve. The curve is planar. We similarly define the twist of a frame by replacing the normal vector by the second director of the frame and the binormal vector by the third director in Equation (2.1).

For the Bishop frame to be twist-free, an initial frame $(\mathbf{t}(0), \underline{\mathbf{d}}_2(0), \underline{\mathbf{d}}_3(0))$ is parallel-transported along the curve. The second vector of the transported frame at each arc-length parameter s is updated tangent to the curve. The frame remains orthogonal along the curve, which defines the third vector as the cross-product of the tangent and the second vector. The parallel transport of a vector $\mathbf{v}$ orthogonal to the tangent of the curve is given by

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}s} \cdot (\mathbf{t} \times \mathbf{v}) = 0,$$

which uniquely defines a vector field along the curve given an initial vector at $s = 0$. Integrating the above for $\underline{\mathbf{d}}_2$ and defining $\underline{\mathbf{d}}_3 = \mathbf{t} \times \underline{\mathbf{d}}_2$ gives the Bishop frame.

By construction of the Bishop frame, the geometric torsion in Equation (2.1) vanishes. The Bishop frame is a natural choice for the description of the deformation of elastic curves.

Discrete Counterparts

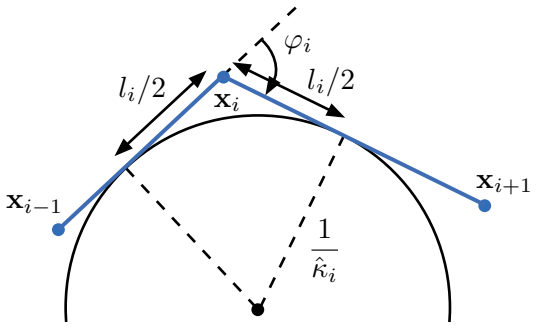
To be numerically simulated, the above continuous curve is discretized into a sequence of points $\mathbf{x}_i = \tilde{\gamma}(s_i)$ for some arc-length parameters $(s_i)_{i\in\{1,\dots,n\}}$. The discrete tangent and binormal vectors are given by

$$\mathbf{t}^i = \frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{l^i},\ \mathbf{b}_i = \frac{\mathbf{t}^{i-1} \times \mathbf{t}^i}{|\sin(\varphi_i)|},\ \text{with } l^i = \|\mathbf{x}_{i+1} - \mathbf{x}_i\| \text{ and } \varphi_i = \angle(\mathbf{t}^{i-1}, \mathbf{t}^i),$$

where subscripts indicate vertex quantities and superscripts indicate edge quantities, $\angle(\cdot,\cdot)$ denotes the angle between two vectors. The length of each edge is l^i and the turning angle between two consecutive edges is written φ_i. The length of the Voronoi cell associated with each vertex is $l_i = (l^{i-1} + l^i)/2$.

The discrete pointwise curvature $\hat{\kappa}_i$ is defined as the reciprocal of the radius of the osculating circle of the curve at a vertex i as shown in Figure 2.2. The integrated curvature is then equal to $\kappa_i = l_i\hat{\kappa}_i = 2\tan(\varphi_i/2)$.

The Bishop frame per edge $(\mathbf{t}^i, \underline{\mathbf{d}}_2^i, \underline{\mathbf{d}}_3^i)$ requires a discrete version of the parallel transport. An initial director is usually provided at one end of the discrete curve $\underline{\mathbf{d}}_2^0$, we simply need to provide a way to update the director from one edge to the next. The discrete parallel transport operator



$$\hat{\kappa}_i = \frac{2}{l_i} \tan\left(\frac{\varphi_i}{2}\right)$$

Figure 2.2: Definition of the discrete geometric curvature based on the radius of the osculating circle in analogy to the continuous setting.

$\mathbf{P}^{\mathbf{t}^i}_{\mathbf{t}^{i-1}}$ is a rotation around the binormal vector $\mathbf{b}_i$ that maps the previous tangent vector $\mathbf{t}^{i-1}$ to $\mathbf{t}^i$. Using the turning angle defined above, the discrete parallel transport operator is given by

$$\mathbf{P}^{\mathbf{t}^i}_{\mathbf{t}^{i-1}} = R(\mathbf{b}_i, \varphi_i).$$

The recurrence relation for the second director reads $\underline{\mathbf{d}}^i_2 = \mathbf{P}^{\mathbf{t}^i}_{\mathbf{t}^{i-1}} \underline{\mathbf{d}}^{i-1}_2$, and the third director is given by $\underline{\mathbf{d}}^i_3 = \mathbf{t}^i \times \underline{\mathbf{d}}^i_2$.

Discrete Elastic Energy

The elastic energy of a discrete curve is defined as the sum of stretching, bending, and twisting energies. Each term requires distinguishing between geometric quantities defined for the rest configuration and the deformed configuration as shown on the left in Figure 2.3. The former is denoted by a bar, e.g., $\bar{\kappa}_i$ is the curvature of the rest configuration.

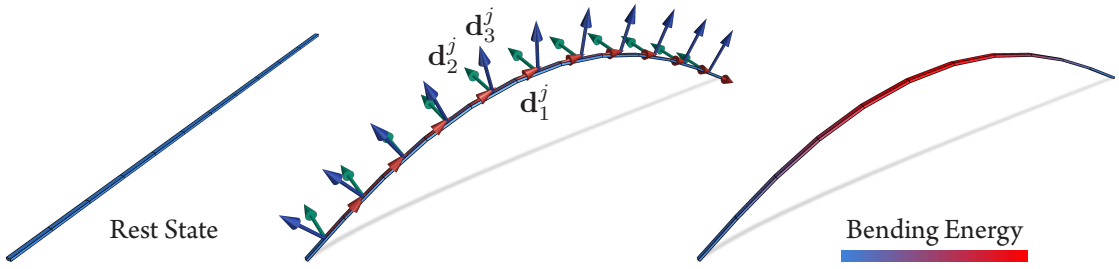


Figure 2.3: A discretized curve (left) and its associated material frame (middle). The total energy of the rod can be computed by summing the stretching, bending, and twisting energies (right).

Stretching scales quadratically with the edge-based axial strain $\epsilon^i = l^i/\bar{l}^i - 1$ as

$$\mathcal{E}_{\text{stretch}} = \frac{1}{2}\sum_{i=1}^{n-1} k_{\text{stretch}} \left(\epsilon^i\right)^2 \bar{l}^i, \tag{2.2}$$

where $k_{\text{stretch}} = EA$ is the stretching stiffness of a material assumed homogeneous with Young's modulus E and cross-sectional area A.

The bending energy requires the introduction of a material frame per edge in the rest state $(\bar{\mathbf{t}}^i, \bar{\mathbf{d}}_2^i, \bar{\mathbf{d}}_3^i)$ and in the deformed state. As opposed to the Bishop frame, the material frame may have twist as it represents the actual orientation of the cross-section of the beam. Twist in the rest state $\bar{m}_i$ is defined at each interior vertex and is typically set to 0 in our setting. Given an initial material director $\bar{\mathbf{d}}_2^0$, the material frame is constructed by first parallel transporting the previous director and rotating it by the twist angle about the next tangent vector. The update rule is then given by

$$\bar{\mathbf{d}}_2^i = R(\bar{\mathbf{t}}^i, \bar{m}_i)\mathbf{P}_{\bar{\mathbf{t}}^{i-1}}^{\bar{\mathbf{t}}^i}\bar{\mathbf{d}}_2^{i-1}, \text{ and } \bar{\mathbf{d}}_3^i = \bar{\mathbf{t}}^i \times \bar{\mathbf{d}}_2^i.$$

For efficiency reason described in [Bergou et al., 2010], the material frame is defined by parallel transporting tangent vectors between the rest and deformed configurations as

$$\mathbf{d}_2^i = \mathbf{P}_{\mathbf{t}^i}^{\mathbf{t}^i}\bar{\mathbf{d}}_2^i, \text{ and } \mathbf{d}_3^i = \mathbf{t}^i \times \mathbf{d}_2^i.$$

We define the discrete integrated curvature vector at a vertex i as

$$\boldsymbol{\kappa}_i = \frac{\kappa_i}{2}\begin{bmatrix} \mathbf{b}_i \cdot \left(\mathbf{d}_3^{i-1} + \mathbf{d}_3^i\right) \\ -\mathbf{b}_i \cdot \left(\mathbf{d}_2^{i-1} + \mathbf{d}_2^i\right) \end{bmatrix},$$

which can be seen as the discrete counterpart of the integrated curvature vector

$$\boldsymbol{\kappa} := \begin{bmatrix} \mathbf{t}' \cdot \mathbf{d}_2 \\ \mathbf{t}' \cdot \mathbf{d}_3 \end{bmatrix} = \begin{bmatrix} \kappa\mathbf{b} \cdot \mathbf{d}_3 \\ -\kappa\mathbf{b} \cdot \mathbf{d}_2 \end{bmatrix},$$

in the continuous setting, where we omitted dependencies to the arc-length parameter s, and we used the triple product rule on $\kappa\mathbf{b} = \mathbf{t} \times \mathbf{t}'$. The bending energy is then defined as the sum of the squared difference between the discrete integrated curvature and the rest integrated curvature as

$$\mathcal{E}_{\text{bend}} = \frac{1}{2}\sum_{i=2}^{n-1} \frac{1}{\bar{l}_i}\left(\boldsymbol{\kappa}_i - \bar{\boldsymbol{\kappa}}_i\right)^\top \mathbf{B}\left(\boldsymbol{\kappa}_i - \bar{\boldsymbol{\kappa}}_i\right),$$

where $\mathbf{B}$ is the anisotropic bending stiffness of the material. Typically, our beams are assumed to be rectangular in cross-section with width and height given by a and b. This is motivated by fabricability constraints. The bending stiffness is then given by

$$\mathbf{B} = E \begin{bmatrix} \frac{ab^3}{12} & 0 \\ 0 & \frac{a^3 b}{12} \end{bmatrix},$$

where the matrix is the area moment of inertia of the cross-section. We visualize the bending energy on a coarsely discretized rectangular beam in Figure 2.3.

The material director orientations are used to define the twisting energy of an elastic rod. For each edge, the directors $(\mathbf{d}_2^i, \mathbf{d}_3^i)$ are decomposed using the directors of the twist-free Bishop frame $(\underline{\mathbf{d}}_2^i, \underline{\mathbf{d}}_3^i)$ as shown in Figure 2.4. The twist angle θ^i is then defined as the angle between the material director $\mathbf{d}_2^i$ and the twist-free director $\underline{\mathbf{d}}_2^i$. The twist in the deformed configuration is then given by

$$m_i = \theta^i - \theta^{i-1} + \bar{m}_i,$$

where $\bar{m}_i$ is the rest twist angle in the rod, which we allow to be non-zero for rods with twisted cross-sections. The rods considered in this thesis have vanishing rest twist.

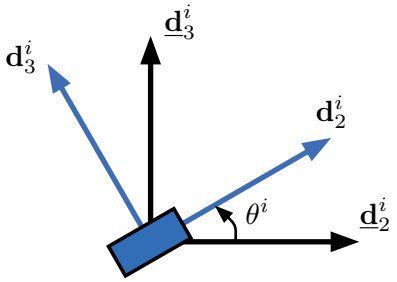


Figure 2.4: Material frame per edge $(\mathbf{d}_2^i, \mathbf{d}_3^i)$ decomposed using the twist-free Bishop frame $(\underline{\mathbf{d}}_2^i, \underline{\mathbf{d}}_3^i)$.

The twisting energy is then defined as the sum of the squared difference between the twist and the rest twist as

$$\mathcal{E}_{\text{twist}} = \frac{1}{2} \sum_{i=1}^{n-1} \frac{k_{\text{twist}}}{\bar{l}_i} (m_i - \bar{m}_i)^2,$$

where $k_{\text{twist}} = G \int_\Omega \left\| (-y, x)^\top + \nabla \psi \right\|^2 \mathrm{d}A$ is the twisting stiffness defined as in [Landau et al., 2020]. G is the shear modulus and ψ is a scalar field on the cross-section Ω that represents the out-of-plane displacement under an applied torsion. It satisfies the Laplace equation $\Delta \psi = 0$

on Ω and the boundary condition $\nabla\psi \cdot \mathbf{n} = (y, -x)^\top \cdot \mathbf{n}$ on the boundary of the cross-section of the beam. The total energy of an elastic rod is then defined as

$$\mathcal{E}_{\text{rod}} = \mathcal{E}_{\text{stretch}} + \mathcal{E}_{\text{bend}} + \mathcal{E}_{\text{twist}}.$$

Equilibrium

An elastic rod is characterized by its segment rest lengths $\bar{l}^i$, its integrated rest curvature $\bar{\boldsymbol{\kappa}}_i$ expressed in the reference material frame, and its twist $\bar{m}_i$. While the latter is set to 0 for fabricability considerations, the others are gathered in the rest state variables $\bar{\mathbf{p}}_{\text{rod}}$. These can typically be extracted from a fabrication state $\bar{\mathbf{x}}_{\text{rod}}$ composed of the vertex positions only, since rest twist is assumed to be vanishing. In general, the deformed state $\mathbf{x}_{\text{rod}}$ of an elastic rod also requires the material frame per edge, which is given by the twist angle θ^i per edge.

Any deformed state of the rod $\mathbf{x}_{\text{rod}}$ increases the elastic energy $\mathcal{E}_{\text{rod}}(\mathbf{x}_{\text{rod}}, \bar{\mathbf{p}}_{\text{rod}}) \geq 0$, and we have $\mathcal{E}_{\text{rod}}(\bar{\mathbf{x}}_{\text{rod}}, \bar{\mathbf{p}}_{\text{rod}}) = 0$. As a global minimizer of the smooth elastic energy, the rest state $\bar{\mathbf{x}}_{\text{rod}}$ is an equilibrium state of the rod. The first-order optimality condition indeed gives

$$\mathbf{f}_{\text{rod}}(\bar{\mathbf{x}}_{\text{rod}}, \bar{\mathbf{p}}_{\text{rod}}) = -\left.\frac{\partial \mathcal{E}_{\text{rod}}}{\partial \mathbf{x}}\right|^\top_{\bar{\mathbf{x}}_{\text{rod}}, \bar{\mathbf{p}}_{\text{rod}}} = 0,$$

where $\mathbf{f}_{\text{rod}}$ gathers the forces exerted on vertices and the torques applied to edges along their tangents.

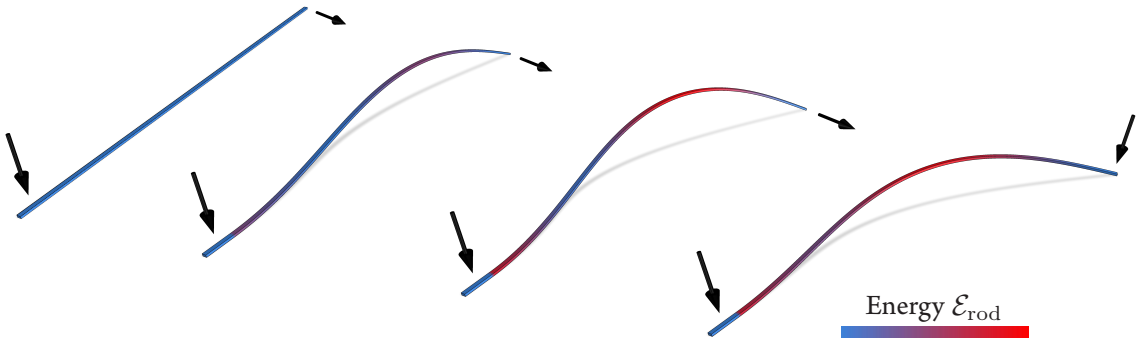


Figure 2.5: Sequence of equilibria $\mathbf{x}^*_{\text{rod}}(\mathbf{x}_{\text{pin}})$ for a straight rod under smoothly varying external constraints $\mathbf{x}_{\text{pin}}$. The black arrows indicate vertex manipulation: The vertex on the left-hand side of the beam stays at a fixed location and the vertex on the right-hand side is gradually slid along the corresponding arrow.

For the rod to settle in a different equilibrium, external forces need to be applied to the rod. In Figure 2.5, we consider the case when all coordinates of a subset of the rod's vertices are pinned to

predefined positions. The equilibrium state $\mathbf{x}^*_{\text{rod}}$ is then defined as the minimizer of the total elastic energy under pinning constraints

$$\mathbf{x}^*_{\text{rod}} = \underset{\mathbf{x}_{\text{rod}}}{\operatorname{argmin}}\, \mathcal{E}_{\text{rod}}(\mathbf{x}_{\text{rod}}, \bar{\mathbf{p}}_{\text{rod}})$$
$$\text{s.t. } \mathbf{S}_{\text{pin}}\mathbf{x}_{\text{rod}} = \mathbf{x}_{\text{pin}},$$

where $\mathbf{S}_{\text{pin}}$ is a selection matrix that extracts the pinned coordinates from the rod's vertices, and $\mathbf{x}_{\text{pin}}$ is the pinned position of the rod's vertices. The first-order Karush-Kuhn-Tucker (KKT) conditions for the optimization problem give

$$0 = -\left.\frac{\partial \mathcal{E}_{\text{rod}}}{\partial \mathbf{x}}\right|^{\top}_{\mathbf{x}^*_{\text{rod}}, \bar{\mathbf{p}}_{\text{rod}}} - \mathbf{S}^{\top}_{\text{pin}}\boldsymbol{\lambda} = \mathbf{f}_{\text{rod}}(\mathbf{x}^*_{\text{rod}}, \bar{\mathbf{p}}_{\text{rod}}) + \mathbf{f}_{\text{ext}}(\bar{\mathbf{p}}_{\text{rod}}, \mathbf{x}_{\text{pin}}),$$

where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers associated with the pinning constraints. The second term regroups the external forces that need to be exerted at the pinned vertices to maintain equilibrium: It has vanishing components except at pinned vertices where it may take non-zero values.

2.2 Deployable Elastic Gridshells

While elastic rods present surprisingly vast and complex behavior [Vidulis et al., 2023, Hafner and Bickel, 2023], our work focuses on the simulation and design of assemblies of elastic rods in the form of gridshells. We introduce the concept of a gridshell and define the total energy of such systems.

Gridshell Assembly

A gridshell consists in an assembly of multiple elastic rods connected through single-axis rotational joints as shown in Figure 2.6. Such joints are assumed to be perfectly rigid and store no elastic energy. They are located at the center of an edge of a discrete elastic rod. The deformed state of a gridshell cannot be directly represented as a concatenation of vertex positions alone since positional constraints and constraints on the orientation of material directors must be satisfied.

These holonomic constraints can be directly enforced through the introduction of deformed state variables for a constrained rod r, $\mathbf{x}_r$, and for a rotational joint j, $\mathbf{x}_j$. The former contains the positions of the rod's vertices and twist angles, whenever the vertex or the angle is not part of an edge containing a joint. The latter contains the position of the joints $\mathbf{j}$ and the orientation ω

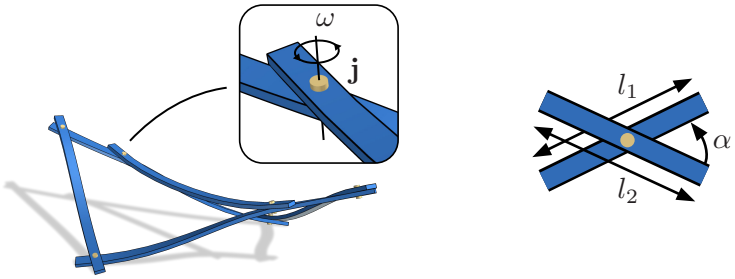


Figure 2.6: Single-axis rotational joint's parameterization. A joint constrains the position and orientation of two edges from two different rods.

of the joint normals, which also determines the orientation of the material directors of the rod edges connected to the joint. Additionally, the vertices adjacent to the joints are computed from the opening angles α and the two edge lengths l_1 and l_2 as shown in Figure 2.6. The deformed state $\mathbf{x}$ of a gridshell is then given by the concatenation of the deformed states of the rods and the joints. The rest state $\mathbf{p}$ of the full gridshell is defined as the concatenation of the rest states of the rods.

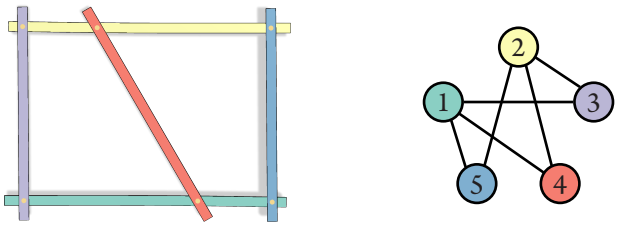


Figure 2.7: Rods connectivity graph. The rods can be grouped in two disjoint families $(1, 2)$ and $(3, 4, 5)$ thanks to the bipartiteness property of the connectivity graph.

A crucial restriction we impose on the rods' connectivity is that they should be split in two disjoint families. In practice this can be verified by making sure the graph of rods connectivity is bipartite as in Figure 2.7. Vertices of the graph represent rods and edges are rotational joints connecting rods. Bipartiteness can be checked through vertex coloring during spanning tree traversal. This bipartite property is essential for the deployment of the gridshell as we will see later. We further impose each joint to connect exactly two rods, one of each family.

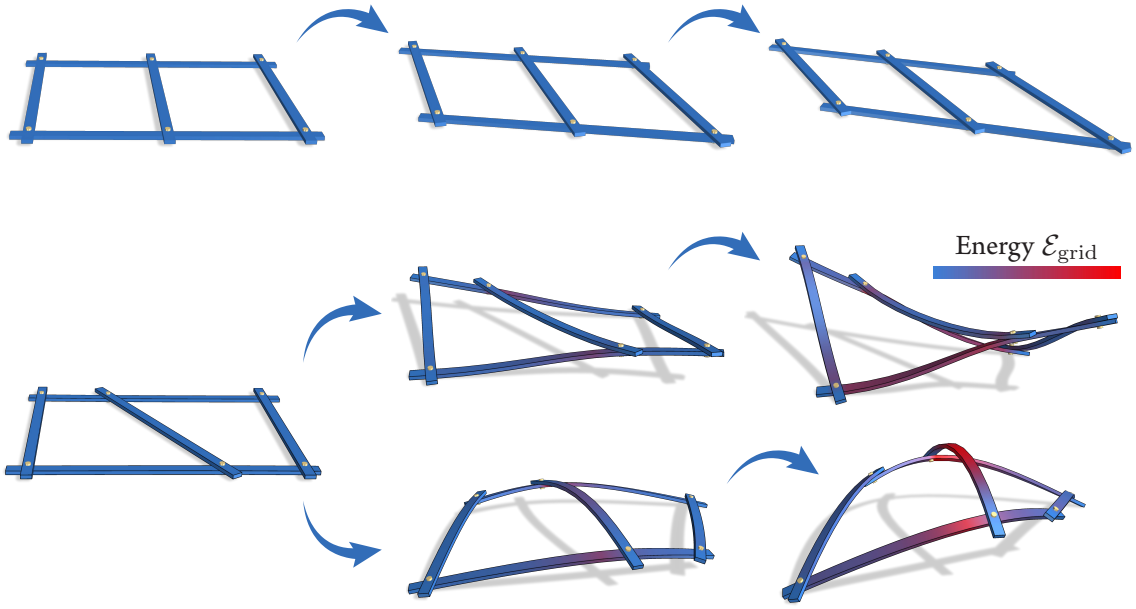


Figure 2.8: Sequences of actuated linkages. While a compatible linkage remains in the plane of the gridshell (top), an incompatible linkage buckles out of plane through uniform torque actuation at the joints (bottom).

Gridshell Deployment

Similar to the case of a single rod, the equilibrium state of a gridshell is defined as the minimizer of the total elastic energy under external constraints. The total energy of a gridshell is defined as the sum of the elastic energies of the rods

$$\mathcal{E}_{\text{grid}}(\mathbf{x}, \mathbf{p}) = \sum_r \mathcal{E}_{\text{rod}}(\mathbf{x}_r, \mathbf{p}_r).$$

The equilibrium state $\mathbf{x}^*$ of a gridshell is then defined as

$$\begin{aligned} \mathbf{x}^* &= \underset{\mathbf{x}}{\operatorname{argmin}}\, \mathcal{E}_{\text{grid}}(\mathbf{x}, \mathbf{p}) \\ &\text{s.t. } a(\mathbf{x}) = \mathbf{a}_{\text{pin}}, \end{aligned} \tag{2.3}$$

where a extracts constrained quantities from the deformed state of the gridshell to model its actuation, and $\mathbf{a}_{\text{pin}}$ is the vector of all the pinned quantities of the gridshell's vertices. While many actuation strategies can be considered e.g., through string actuation or vertex pinning, we focus on torque actuation at the joints. The actuation is given by the average opening angle at the joints $a(\mathbf{x}) = \frac{1}{n_j}\sum_j \alpha_j = \mathbf{e}_\alpha^\top \mathbf{x}$ and the pinned variable is written $\bar{\alpha}_{\text{pin}}$. The total number of joints

is denoted by n_j and $\mathbf{e}_\alpha$ selects the opening angles from the deformed state of the gridshell and scales them by the number of joints.

Expressing the KKT conditions of the optimization problem gives

$$\begin{cases} -\left.\dfrac{\partial \mathcal{E}_{\text{grid}}}{\partial \mathbf{x}}\right|^{\top}_{\mathbf{x}^*,\mathbf{p}} - \tau \mathbf{e}_\alpha = 0 \\ \mathbf{e}_\alpha^{\top}\mathbf{x}^* = \bar{\alpha}_{\text{pin}}, \end{cases} \tag{2.4}$$

where τ is the Lagrange multiplier associated with the actuation constraints. The second term is the *uniform* torque exerted at the joints to maintain equilibrium.

Depending on the rest variables $\mathbf{p}$, the gridshell may be geometrically incompatible as in Figure 2.8. The top row shows a compatible linkage, where a pantograph motion can be observed: The elastic beams do not deform during actuation. If the rest lengths are instead set to be incompatible, torque actuation forces the beams to deform and the whole assembly to buckle out of plane as shown in the bottom two rows. Buckling is favored by the orientation of the weak axis of the beams, which is orthogonal to the plane of the gridshell. These gridshells also enjoy a quasi-mirror symmetry about the plane of the gridshells (up to the z-offset between the two families of beams). In practice, this symmetry is broken by a prescribed perturbation of the vertex positions as well as imperfections in the fabrication process and the gridshell may deploy towards one of the two mirrored buckled states.

Geometry of Gridshells

Our first geometric observation came from the notion of compatibility of a linkage. However, this merely allows predicting whether a gridshell will deploy or not. One may observe in Figure 2.8 that the surface approximated by the deployed gridshell may have positive or negative Gaussian curvature depending on the sign of variation of the average opening angle. More precisely, the Gauss-Bonnet theorem, stated below and illustrated in Figure 2.9, provides the connection between opening angles and Gaussian curvature.

Theorem 1 (Gauss-Bonnet Theorem)

Let M be a compact two-dimensional Riemannian manifold with closed boundary $\partial M = \bigcup_{i=1}^{n} C_i$ for some closed, piecewise, simple, and regular curves $C_1, \ldots, C_n$. Let K be the Gaussian curvature of the surface patch, κ_g be the geodesic curvature of each curve, β_i be the turning angles where pairs of curves meet, and $\chi(M)$ be the Euler characteristic of M. Then

$$\int_M K \mathrm{d}A + \sum_{i=1}^{n-1} \int_{C_i} \kappa_g \mathrm{d}s + \sum_{i=1}^{n} \beta_i = 2\pi\chi(M).$$

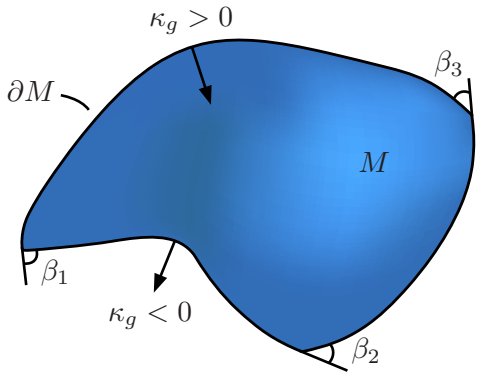


Figure 2.9: The Gauss-Bonnet theorem relates curvature and topology.

Most of the surfaces considered in this thesis are homotopic to a disk and the Euler characteristic is a topological invariant, we have $\chi(M) = 1$. Applying the Gauss-Bonnet theorem to one of the two quadrilaterals of the rest state in Figure 2.10 simply gives that the sum of internal angles equals 2π.

In the deployed case, we make the simplifying assumption that the geodesic curvature of the beam remains negligible. In practice, this amounts to assuming *ribbon-like* beams, for which twisting and bending along the weak axis of the cross-section is more favorable than bending directly along the strong axis. Geometrically, the beams follow geodesics of the underlying surface. The Gauss-Bonnet theorem simplifies to

$$\int_M K \mathrm{d}A = 2\pi - \sum_{i=1}^{n} \beta_i = \sum_{i=1}^{n} (\bar{\beta}_i - \beta_i),$$

where $\bar{\beta}_i$ (resp. β_i) are the turning angles in the rest (resp. deployed) state, and M is the surface enclosed by the beams. The Gauss-Bonnet theorem then provides a direct connection between the opening angles and the Gaussian curvature of the surface. The angle deficit shown in Figure 2.10 is in qualitative agreement: The surface approximated by the gridshell has negative integrated Gaussian curvature when the angle deficit is positive and vice versa. In this example, the surface is a B-spline surface that interpolates the rods' vertices.

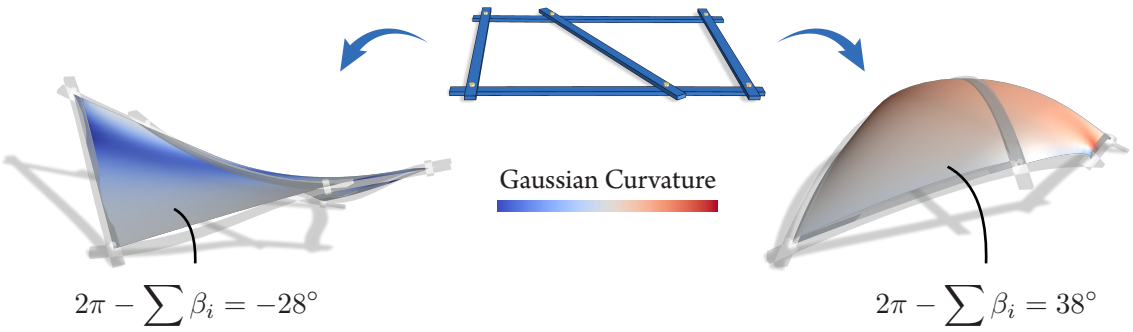


Figure 2.10: A linkage can be deployed by either decreasing or increasing the average opening angle $\bar{\alpha}_{\text{pin}}$. The Gaussian curvature becomes negative when the angles' deficit per quadrilateral is negative and vice versa.

However, this geometric approach has at least two caveats. First, our assumption on the vanishing geodesic curvature of the elements in the deployed state may not hold exactly. Some Gaussian curvature may be traded for geodesic curvature in the Gauss-Bonnet theorem, the amount of which is dictated by the cross-section of the beams. Second, the theorem only informs about the *integrated* Gaussian curvature over each quadrilateral and fails to prescribe the exact shape of the surface. Fitting a surface to the gridshell is in fact an-ill posed problem and can be addressed using the inverse design framework we introduce next.

2.3 Inverse Design of Deployable Elastic Gridshells

The design of deployable gridshells is a challenging task due to the complex behavior of assemblies of elastic beams. The intricate connection between the linkage's rest state and its actuated state is a major obstacle to the manual design of such structures. We introduce the concept of an optimization-based approach to design the geometry of the gridshell elements.

Optimization Problem

The design of a gridshell is cast as an optimization problem where the objective is to minimize a quality metric of the actuated linkage $\mathcal{J}$. Our metric is split into two terms (see Figure 2.11): The first term measures the amount of energy stored in the deployed assembly $\mathcal{E}_{\text{grid}}$, and the second term quantifies the deviation of the gridshell from a target shape $\mathcal{T}$ e.g., by computing the squared distance of each joint to their projections on the surface. In this didactic example, we restrict the design space of assemblies to two parameters: The incompatibility offset δ_x (see Figure 2.11) and the average opening angle $\bar{\alpha}_{\text{pin}}$. The design variables are stored in $\mathbf{d}$.

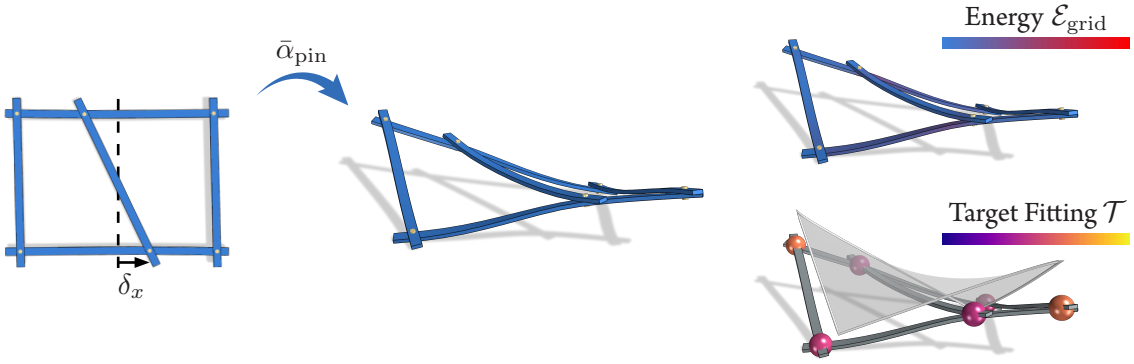


Figure 2.11: Input gridshell design for the inverse problem. The linkage is parameterized by an incompatibility offset δ_x and the average opening angle in the deployed state $\bar{\alpha}_{\text{pin}}$. The objective is split between the elastic energy in the deployed state $\mathcal{E}_{\text{grid}}$ and the deviation from a target shape $\mathcal{T}$.

The objective is measured at the deployed state $\mathbf{x}^*(\mathbf{d})$ of the gridshell defined in Equation (2.3) and for rest quantities $\mathbf{p}(\mathbf{d})$ extracted from the design variables. The optimization problem is then defined as

$$\min_{\mathbf{d}} \mathcal{J}(\mathbf{d}) = \mathcal{E}_{\text{grid}}(\mathbf{x}^*(\mathbf{d}), \mathbf{p}(\mathbf{d})) + \mathcal{T}(\mathbf{x}^*(\mathbf{d})), \tag{2.5}$$

where each term is weighted and scaled by appropriate physical and geometric quantities: The elastic energy is scaled by YV with Y the Young's modulus of the rod's material, V the sum of the volume of the beams, and the target shape deviation is scaled by the diagonal of the bounding box of the target surface squared.

Sensitivity Analysis

The minimization problem is solved using a gradient-based optimization algorithm and requires the computation of the gradient of the objective $\mathcal{J}$ with respect to the design variables $\mathbf{d}$. We now

describe the adjoint method we use throughout this thesis, which allows efficiently computing such derivatives.

We start by defining the objective alternatively as $\tilde{\mathcal{J}}(\mathbf{x}, \mathbf{p}) := \mathcal{E}_{\text{grid}}(\mathbf{x}, \mathbf{p}) + \mathcal{T}(\mathbf{x})$ so that $\mathcal{J}(\mathbf{d}) = \tilde{\mathcal{J}}(\mathbf{x}^*(\mathbf{d}), \mathbf{p}(\mathbf{d}))$. The adjoint state method consists in first using the chain rule to write

$$\frac{\partial \mathcal{J}}{\partial \mathbf{d}} = \frac{\partial \tilde{\mathcal{J}}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}^*}{\partial \mathbf{d}} + \frac{\partial \tilde{\mathcal{J}}}{\partial \mathbf{p}} \frac{\partial \mathbf{p}}{\partial \mathbf{d}}.$$

The second term requires the gradient of the energy with respect to the design variables, which can be computed following [Panetta et al., 2019], and the derivative of the rest lengths with respect to the offset parameter δ_x. The latter can straightforwardly be computed through algebraic manipulations. The first term however requires a different treatment.

The gradient of the objective with respect to the deformed state $\mathbf{x}$ is also given following [Panetta et al., 2019], but the equilibrium state is only defined implicitly as the solution of a constrained minimization problem. Taking the KKT conditions in Equation (2.4) and differentiating against the design variables gives

$$\begin{cases} \dfrac{\partial^2 \mathcal{E}_{\text{grid}}}{\partial \mathbf{x}^2} \dfrac{\partial \mathbf{x}^*}{\partial \mathbf{d}} + \dfrac{\partial^2 \mathcal{E}_{\text{grid}}}{\partial \mathbf{x} \partial \mathbf{d}} + \mathbf{e}_\alpha \dfrac{\partial \tau}{\partial \mathbf{d}} = 0 \\ \mathbf{e}_\alpha^\top \dfrac{\partial \mathbf{x}^*}{\partial \mathbf{d}} = \begin{bmatrix} 0 & 1 \end{bmatrix}. \end{cases}$$

Defining $\mathbf{H}_{\text{grid}}$ as the Hessian of the total elastic energy and solving the system for the Jacobian gives

$$\begin{aligned} \frac{\partial \mathbf{x}^*}{\partial \mathbf{d}} &= -\mathbf{H}_{\text{grid}}^{-1} \left(\frac{\partial^2 \mathcal{E}_{\text{grid}}}{\partial \mathbf{x} \partial \mathbf{d}} + \mathbf{e}_\alpha \frac{\partial \tau}{\partial \mathbf{d}} \right) \\ &= -\mathbf{H}_{\text{grid}}^{-1} \left(\frac{\partial^2 \mathcal{E}_{\text{grid}}}{\partial \mathbf{x} \partial \mathbf{d}} - \frac{\mathbf{e}_\alpha \mathbf{e}_\alpha^\top \mathbf{H}_{\text{grid}}^{-1}}{\mathbf{e}_\alpha^\top \mathbf{H}_{\text{grid}}^{-1} \mathbf{e}_\alpha} \frac{\partial^2 \mathcal{E}_{\text{grid}}}{\partial \mathbf{x} \partial \mathbf{d}} - \frac{\mathbf{e}_\alpha \begin{bmatrix} 0 & 1 \end{bmatrix}}{\mathbf{e}_\alpha^\top \mathbf{H}_{\text{grid}}^{-1} \mathbf{e}_\alpha} \right). \end{aligned}$$

While it is perfectly possible to directly compute the Jacobian in this example, the method would poorly scale as the number of design variables in $\mathbf{d}$ increases. We would have to solve a linear system for each design variable, which becomes computationally expensive with the number of variables. The adjoint method allows solving two linear systems no matter the dimensionality of the design space. We write

$$\frac{\partial \tilde{\mathcal{J}}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}^*}{\partial \mathbf{d}} = -\frac{\partial \tilde{\mathcal{J}}}{\partial \mathbf{x}} \left(\mathbf{H}_{\text{grid}}^{-1} - \frac{\mathbf{H}_{\text{grid}}^{-1} \mathbf{e}_\alpha \mathbf{e}_\alpha^\top \mathbf{H}_{\text{grid}}^{-1}}{\mathbf{e}_\alpha^\top \mathbf{H}_{\text{grid}}^{-1} \mathbf{e}_\alpha} \right) \frac{\partial^2 \mathcal{E}_{\text{grid}}}{\partial \mathbf{x} \partial \mathbf{d}} + \frac{\partial \tilde{\mathcal{J}}}{\partial \mathbf{x}} \mathbf{H}_{\text{grid}}^{-1} \frac{\mathbf{e}_\alpha \begin{bmatrix} 0 & 1 \end{bmatrix}}{\mathbf{e}_\alpha^\top \mathbf{H}_{\text{grid}}^{-1} \mathbf{e}_\alpha}.$$

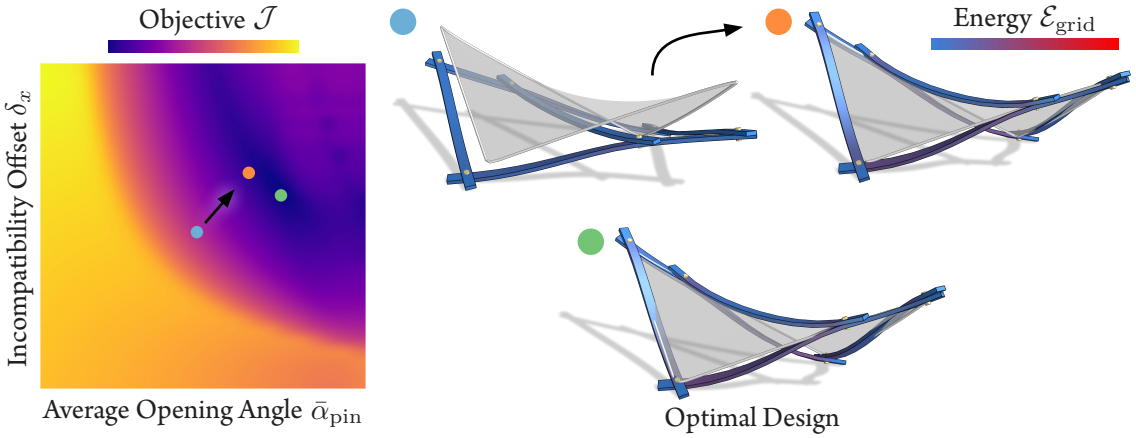


Figure 2.12: Illustrative inverse design problem. The energy landscape is shown on the left, and several states are shown on the right-hand side. The blue state is the initial state, the orange state is the result of one optimization step along a search direction represented by the black arrow. The green state is the global minimum.

We now define the adjoint state vector $\mathbf{v}$ and a vector $\mathbf{w}_\alpha$ as the solutions of the following linear systems

$$\begin{bmatrix} \mathbf{H}_{\text{grid}} & \mathbf{e}_\alpha \\ \mathbf{e}_\alpha^\top & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ v_\tau \end{bmatrix} = \begin{bmatrix} -\dfrac{\partial \tilde{\mathcal{J}}}{\partial \mathbf{x}}^\top \\ 0 \end{bmatrix}, \text{ and } \mathbf{H}_{\text{grid}} \mathbf{w}_\alpha = \mathbf{e}_\alpha,$$

where v_τ is a scalar. The term in the original chain rule can then be written as

$$\frac{\partial \tilde{\mathcal{J}}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}^*}{\partial \mathbf{d}} = \mathbf{v}^\top \frac{\partial^2 \mathcal{E}_{\text{grid}}}{\partial \mathbf{x} \partial \mathbf{d}} - (\mathbf{v} + v_\tau \mathbf{w}_\alpha)^\top \frac{\mathbf{e}_\alpha \begin{bmatrix} 0 & 1 \end{bmatrix}}{\mathbf{w}_\alpha^\top \mathbf{e}_\alpha}.$$

Second-order derivatives of the objective $\mathcal{J}$ can be computed similarly. We leave them aside in this illustrative example and use first-order algorithms to solve the minimization problem formulated in Equation (2.5). Such algorithms usually require an initial state, shown in blue in Figure 2.12, and sequentially update the design variables to converge to a solution. A search direction is needed at each iteration, which is computed based on the objective's gradient $\frac{\partial \mathcal{J}}{\partial \mathbf{d}}$. After one step, the orange design significantly improves on the quality measure. The process is repeated until convergence is reached.

While we cannot ensure that the optimization algorithm converges to the green global minimum, we can guarantee that the solution is a critical point of the objective. A good initialization is then crucial to improve the chances of finding a good solution. Therefore, the initial state is often the

solution to a simpler problem, possibly derived from a geometric abstraction similar to the one exposed in Section 2.2.

Chapter 3

C-shells: Deployable Gridshells with Curved Beams

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.

3.1 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

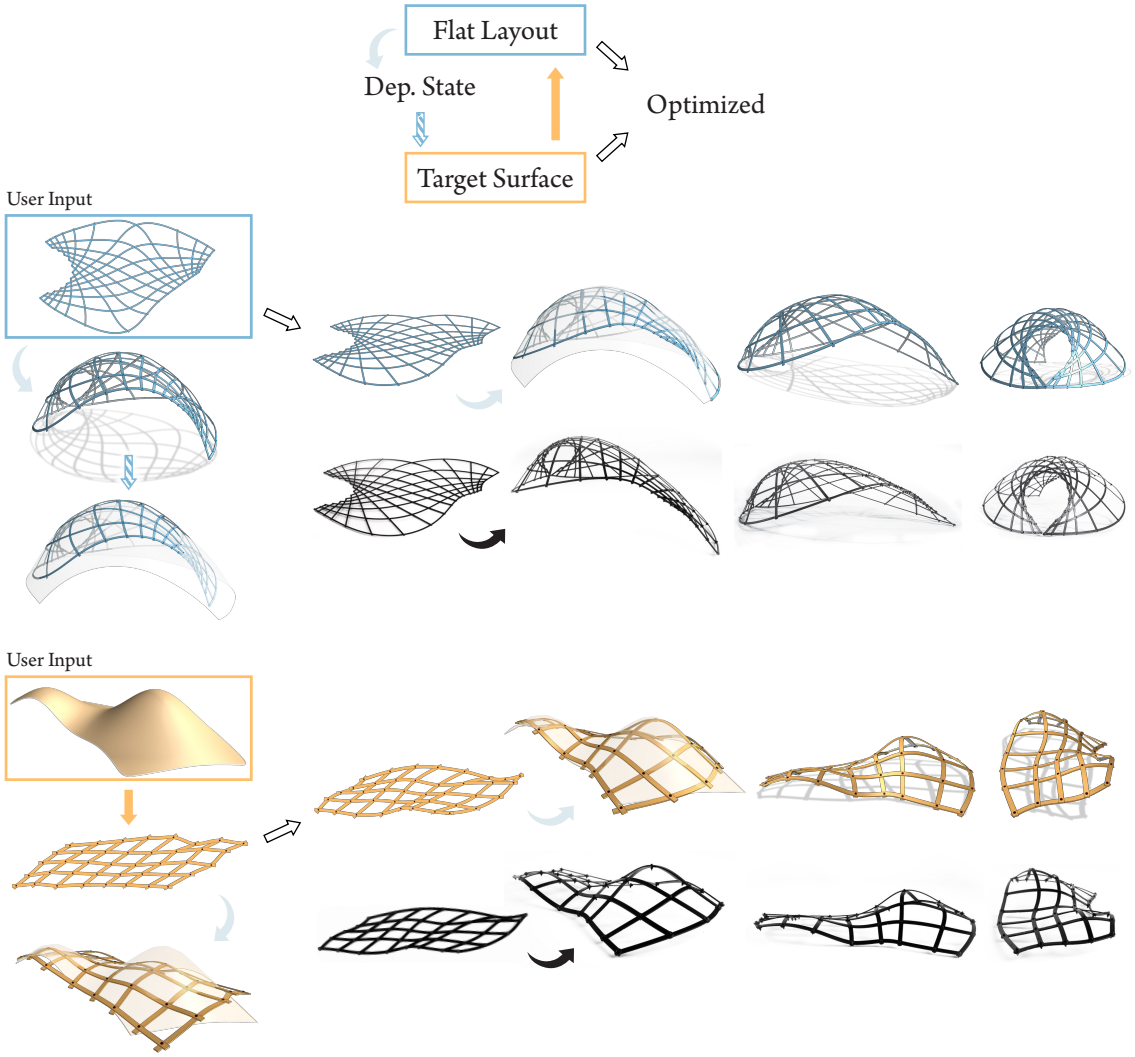


Figure 3.1: Forward exploration (top) and inverse design (bottom) of C-shells. The target design surface is either provided directly by the user or created by deploying an initial beam layout. Subsequent design optimization then solves for the planar curved beam shapes such that the deployed linkage 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.

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 non-uniformity leads to kinetic incompatibilities when the linkage is deployed, which forces the structure to

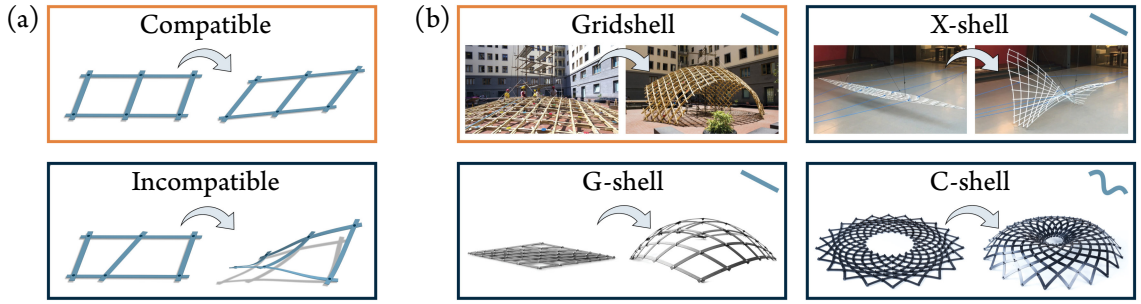


Figure 3.2: Existing deployable beam structures (gridshell, X-shell, G-shell) assemble *straight* beams in a (near) planar configuration that can then be deployed into a desired 3D target surface. Traditional gridshells employ a compatible scissor linkage and are deployed by constraining the boundary. 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.

buckle out of plane to reach an elastic equilibrium (see Figure 3.2). 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 chapter, 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 3.1). 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 3.21).

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 3.15). 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 3.10). 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 kinetically 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 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.

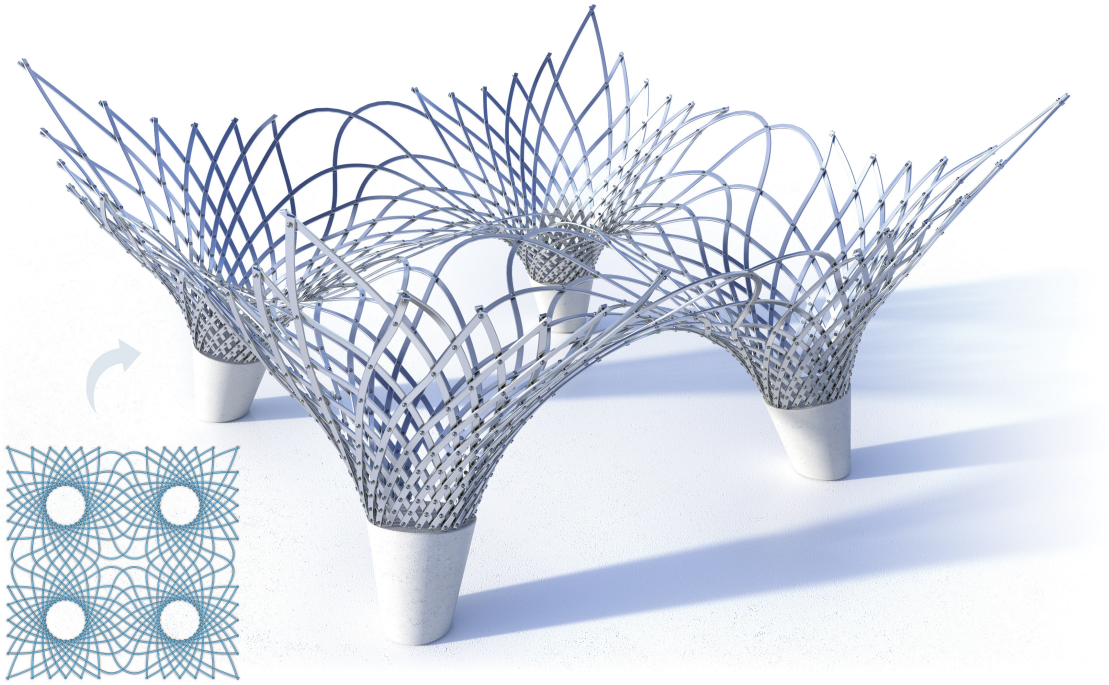


Figure 3.3: A design study for a large-scale architectural application.

- 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.

3.2 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, Pérez et al., 2017, Jourdan et al., 2022], origami [Dudte et al., 2016], auxetic materials [Konaković-Luković et al., 2018, Chen et al., 2021], 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

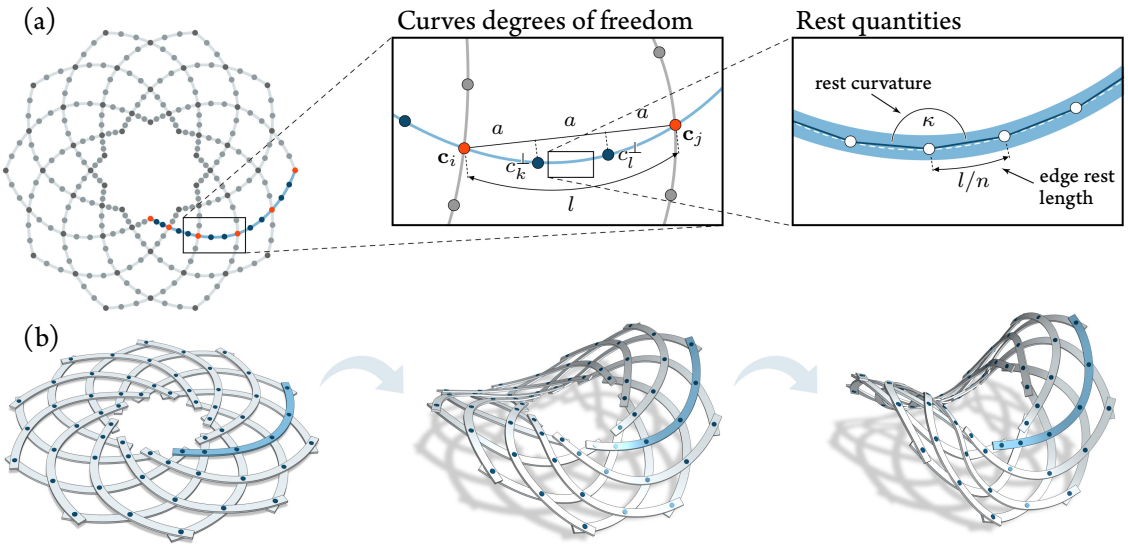


Figure 3.4: C-shell representation. (a) The curve linkage is composed of interpolating splines that meet at joints $\mathbf{c}_i$. Intermediate interpolation points are evenly distributed along the chord joining consecutive joints and are constrained to move orthogonally to it according to scalar parameter $c_k^\perp$. Each curve is discretized into a polyline, from which we extract the discrete curvature κ. The arc length l is evenly distributed among edge 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.

beams connected via rotational joints. We categorize such gridshells into two different classes (Extrinsic and Intrinsic) based on their deployment (Figure 3.2). 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 [Garg et al., 2014, Baek et al., 2018, 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, 2023]. 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 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 et al., 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.

3.3 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.

Spline Representation

To facilitate design optimization, we define an intermediate spline representation for the curved beams that offers two main advantages: (i) A significantly reduced number of optimization vari-

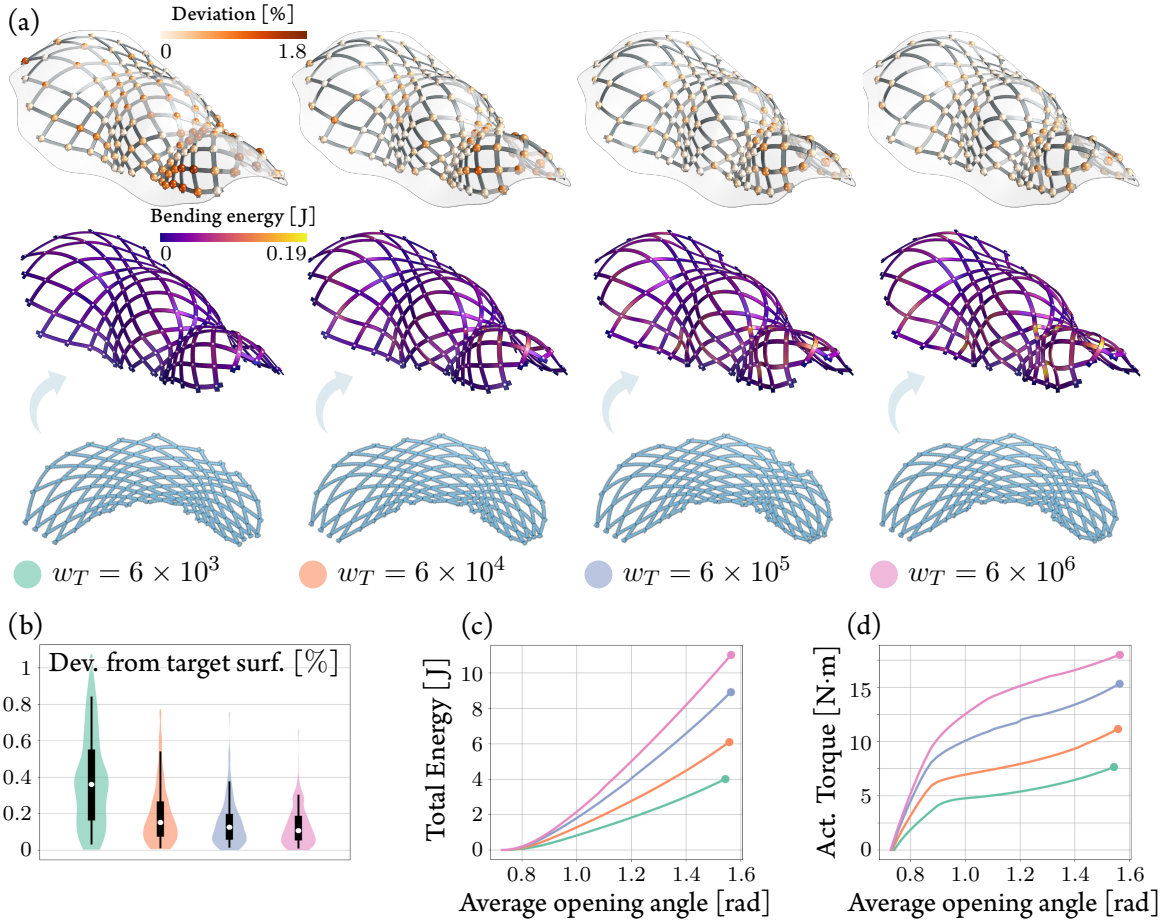


Figure 3.5: (a) The surface fitting weight w_T controls the trade-off between target fitting (top) and elastic energy (middle). Deviation to the target surface is reported as a percentage of the diagonal of the surface's bounding box. (b) Target surface fitting histograms measure the deviation from the 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.

ables, 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 3.4). 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.

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 3.4). 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 3.5) 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.

3.4 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.

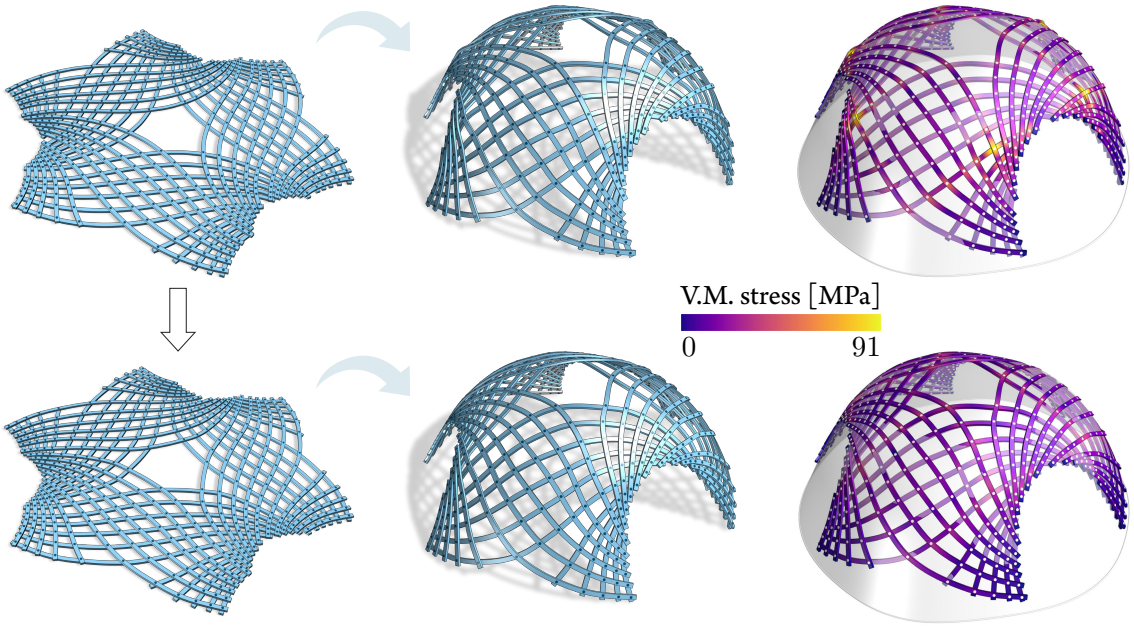


Figure 3.6: Manual design of a C-shell linkage often leads to high stresses in the deployed shape (left). Our optimization reduces stress concentrations without significantly affecting the deployed shape, which helps avoid material failure during deployment (right). Here the maximum von Mises stress is reduced by 78%.

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:

$$\mathbf{x}^*_{\text{3D}}(\mathbf{p}, \bar{\alpha}^{\text{tgt}}) := \underset{\mathbf{x}}{\text{argmin}} \frac{1}{YV_0} E(\mathbf{x}, \mathbf{p}) + T(\mathbf{x}) \qquad (3.1)$$
$$\text{s.t.} \quad \bar{\alpha}(\mathbf{x}) = \bar{\alpha}^{\text{tgt}},$$

where E is the elastic energy summed over all DERs in the linkage, T is the surface-attraction energy defined later in Equation (3.4), 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 $\mathbf{x}^*_{3\mathrm{D}}(\mathbf{p}, \bar{\alpha}^{\mathrm{tgt}} + \Delta\bar{\alpha}) = \mathbf{x}^*_{3\mathrm{D}}(\mathbf{p}, \bar{\alpha}^{\mathrm{tgt}}) + \frac{\partial \mathbf{x}^*_{3\mathrm{D}}}{\partial \bar{\alpha}^{\mathrm{tgt}}}\Delta\bar{\alpha} + O(\Delta\bar{\alpha}^2)$ for the next equilibrium using sensitivity analysis.

One significant deviation from the X-shells simulation framework was needed regarding the constraint $\bar{\alpha}(\mathbf{x}) = \bar{\alpha}^{\mathrm{tgt}}$. 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 Equation (3.1), 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 is 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 $\bar{\alpha}_{\mathbf{x}} = \bar{\alpha}^{\mathrm{tgt}}$ that can be applied by eliminating $\bar{\alpha}_{\mathbf{x}}$ from the 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 Equation (3.1), 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.

Stress Computation

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

3.5 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 3.6). We discuss in Section 3.6 how to obtain the initial beam layout through either forward exploration or inverse design.

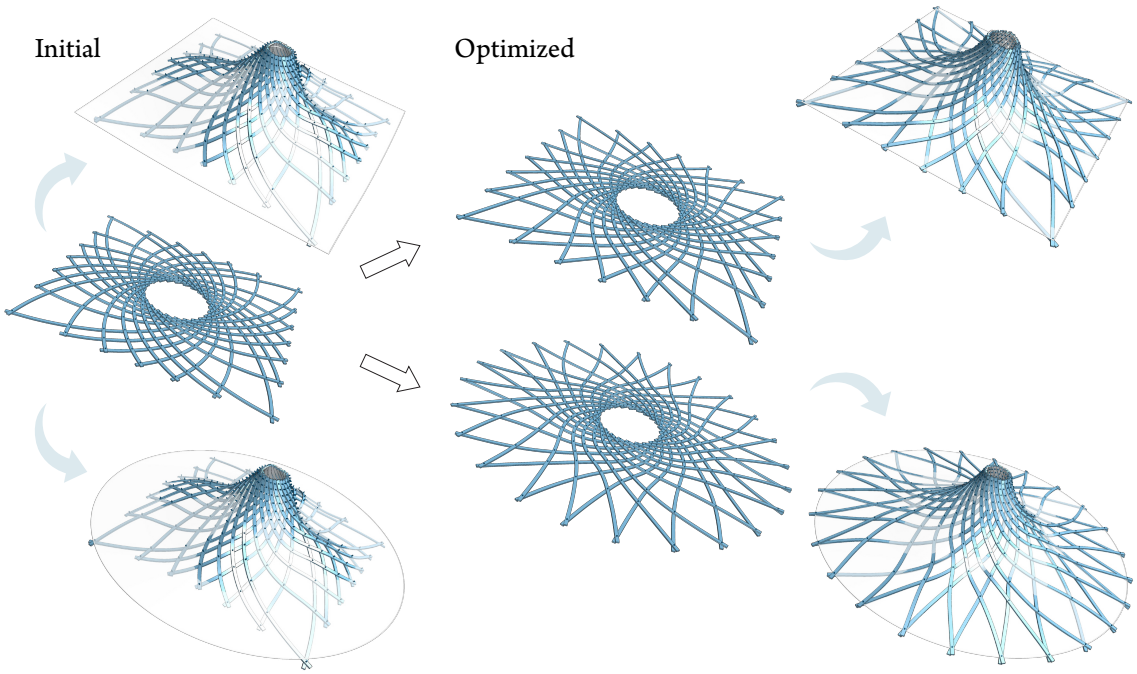


Figure 3.7: Controlling target fitting. Specifying 200 times higher weights for the square (top left) and circle (bottom left) boundary points allows optimizing the initial design (middle left) to accurately reproduce the desired target shapes.

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:

$$J(\mathbf{x}, \mathbf{q}, \mathbf{p}) := \frac{1}{E_0} E(\mathbf{x}) + T(\mathbf{x}) + R(\mathbf{q}, \mathbf{p}), \tag{3.2}$$

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:

$$\begin{aligned} &\underset{\mathbf{q}, \bar{\alpha}^{\mathrm{tgt}}}{\operatorname{argmin}} \; J(\mathbf{x}^*_{\mathrm{3D}}(\mathbf{p}(\mathbf{q}), \bar{\alpha}^{\mathrm{tgt}}), \mathbf{q}, \mathbf{p}(\mathbf{q})) \\ &\text{s.t.} \quad \alpha_{\min}(\mathbf{q}) \geq \epsilon. \end{aligned} \tag{3.3}$$

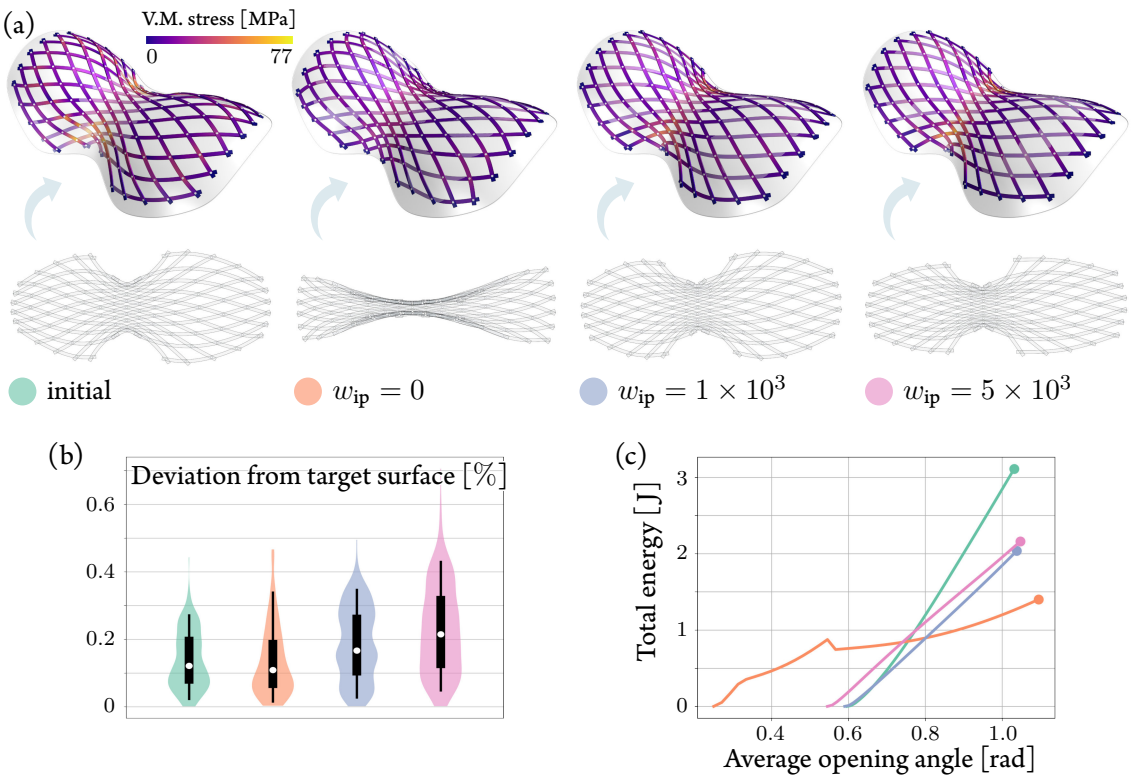


Figure 3.8: Effect of Laplacian regularization on the design optimization. (a) Stress distribution in the deployed state (top) and planar linkage layout. Self-intersections are clearly noticeable without regularization. (b) Target fitting histograms. White dots represent the medians, thick lines show 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.

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 Equation (3.2) 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

$$T(\mathbf{x}) := \frac{w_T}{2l_0^2} \left(\|\mathbf{x} - P_{\text{surf}}(\mathbf{x})\|^2_{W_{\text{surf}}} + \|\mathbf{x} - \mathbf{x}_t\|^2_{W_t} \right), \tag{3.4}$$

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 3.5 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 3.5 indicate, the target-fitting objective saturates at $w_T = 6 \times 10^5$.

Figure 3.7 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 presence of large 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:

$$R(\mathbf{q}, \mathbf{p}) := w_{\text{ip}} R_{\text{ip}}(\mathbf{q}) + w_\kappa R_\kappa(\mathbf{p}). \tag{3.5}$$

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 (3.5).

Figure 3.8 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

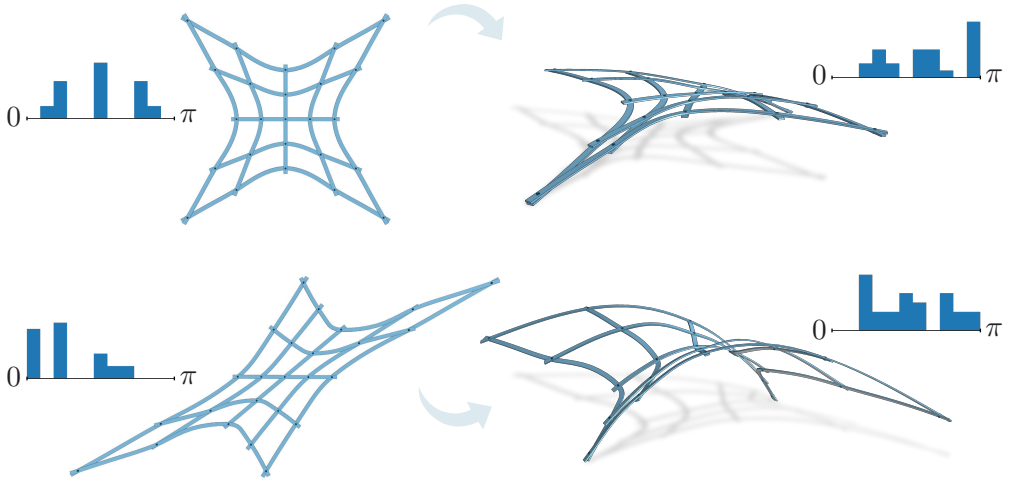


Figure 3.9: Impact of the angle distribution in the flat state regarding achievable deployed shapes. Having angles spanning a larger range in the flat state (top row) does not allow deploying the linkage as much as when angles are more consistently closed or opened (bottom row). The histograms show the distribution of opening angles for each state of each design.

stress, larger total energy, and increased deviation of the deployed structure from the target surface. For further details, please refer to the supplemental material.

3.6 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 3.1. 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.

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

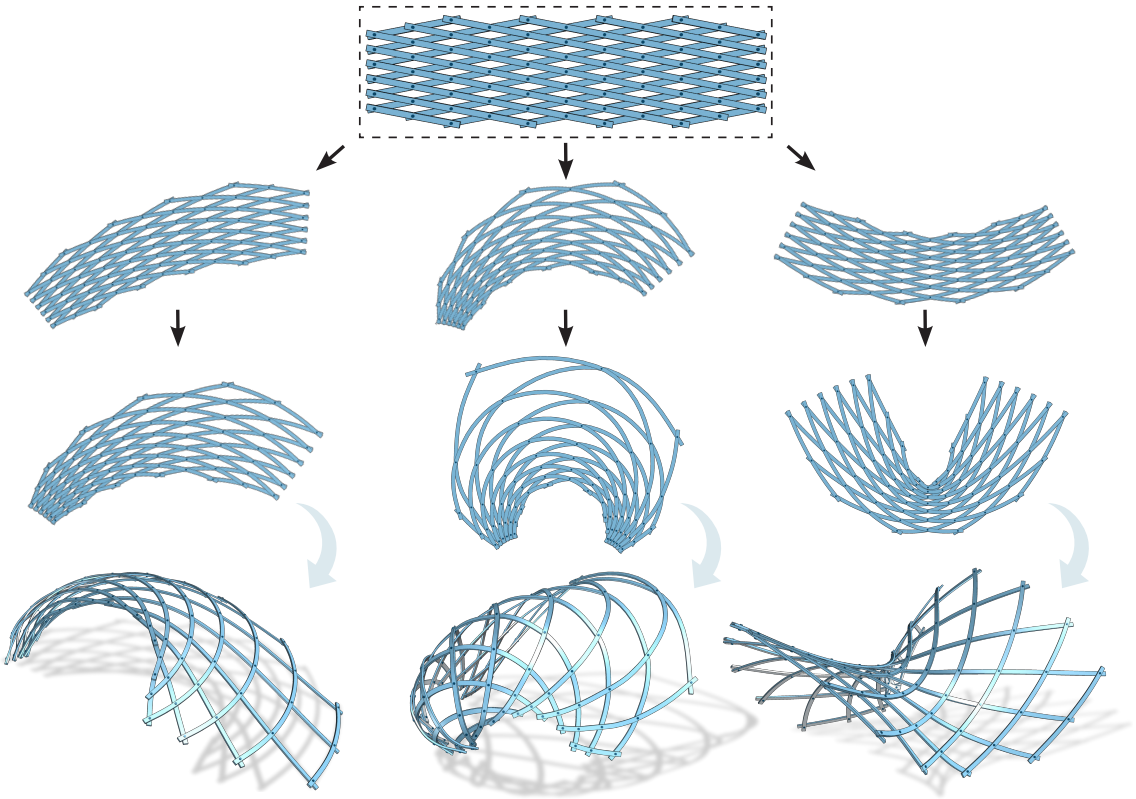


Figure 3.10: Editing beam geometry using conformal deformations in the plane offers great flexibility to adapt an initial regular layout. These deformations preserve the angle distribution at the joints to retain a large deployment range.

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 or decreasing the average angle, respectively) without inverting the layout. As illustrated in Figure 3.9, 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 3.10, conformal maps offer great versatility for transforming straight beam

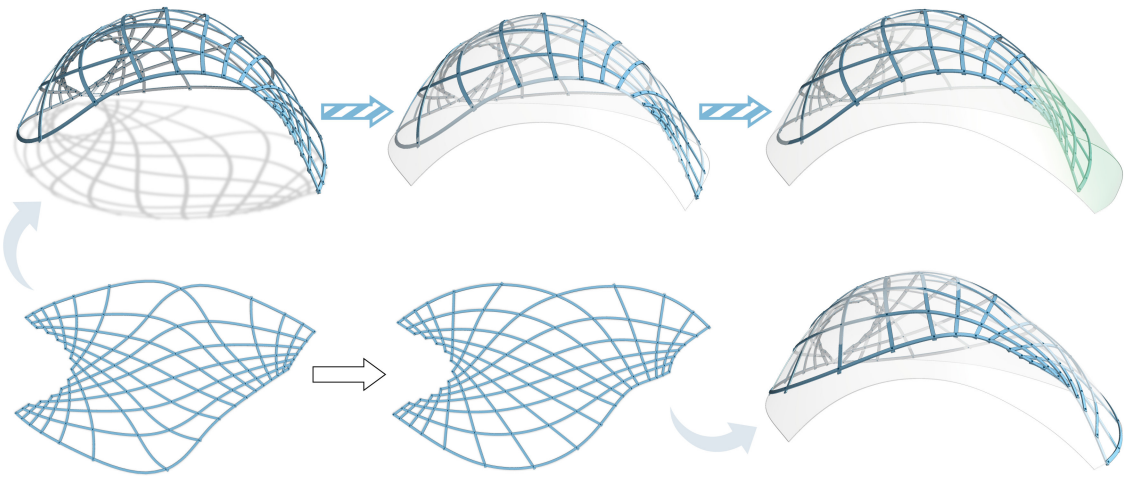


Figure 3.11: C-shell editing. Once an overall satisfactory beam layout and target shape have been found, the C-shell can be further edited by smoothly deforming the target surface. The design optimization then adapts the planar layout accordingly. Regions of larger edits are indicated in green.

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 3.5, 3.8, 3.18, and five examples from Figure 3.21). Additionally, ring-shaped layouts can be designed by parameterizing families of curves and copying them around a circle (see Figure 3.4, 3.7, 3.15, 3.6, 3.16, 3.20). Lastly, different layouts can be composed by cutting and reconnecting their boundary curves as shown in Figure 3.3.

Editing Initial planar layouts created with the forward process are improved by running our design optimization discussed in Section 3.5. 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 Equation (3.4) 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

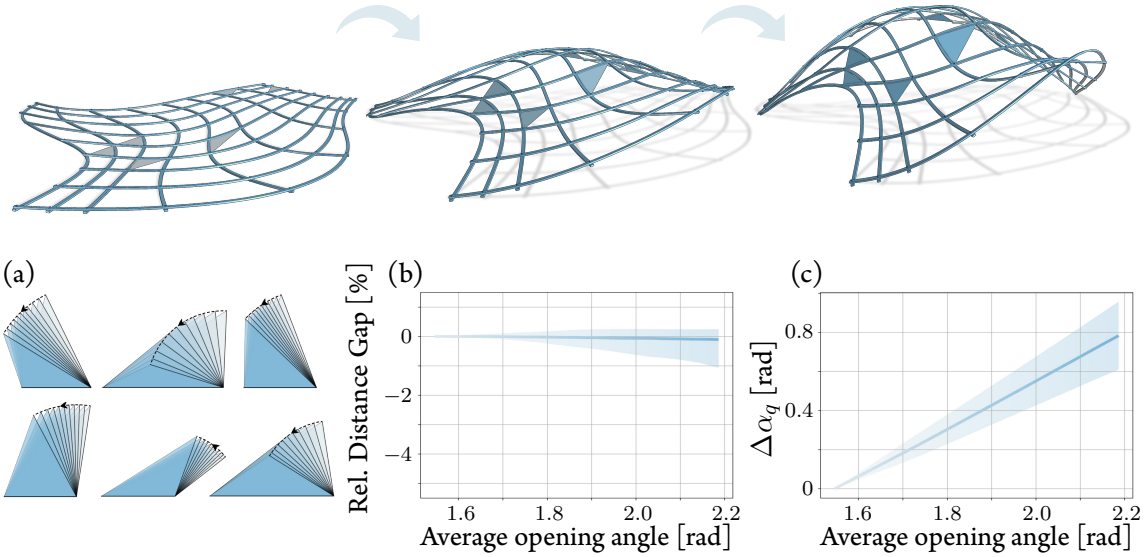


Figure 3.12: C-shell deployment kinetics. (a) Isolated triangles connecting neighboring joints in the linkage reveal that one vertex follows an approximately circular trajectory when the opposite edge is kept fixed during deployment. Inter-joints distances remain approximately constant with a maximum relative change of 1.8% during deployment. (b) median and 5^{th} to 95^{th} percentile range of relative distances. (c) mean and standard deviation of the opening angle increment as the linkage deploys.

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 3.11 shows an example of such an editing operation.

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.

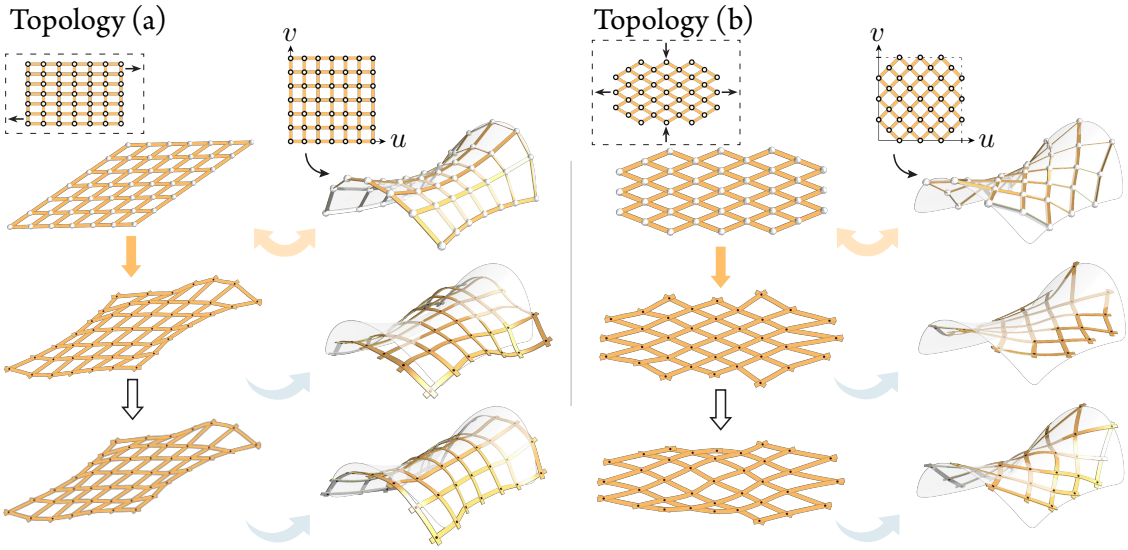


Figure 3.13: Our planarization algorithm applied to different linkage layouts. Topology (a) follows the uv-isolines to trace the target joint positions. Topology (b) follows uv-isolines rotated by $\pi/4$. The flat layouts are initialized by deploying the regular grids as indicated in the dashed boxes 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.

Planarization seeks a minimum of a weighted sum of terms:

$$\begin{aligned} \mathbf{c}^*, \mathbf{u}^*, \mathbf{v}^* := \underset{\mathbf{c},\mathbf{u},\mathbf{v}}{\operatorname{argmin}} \; & \mathcal{E}_{\text{len}} + \mathcal{E}_{\alpha} + \mathcal{E}_{\text{eq}} + \mathcal{E}_{\text{bnd}} \\ \text{s.t. } & \mathcal{C}_{\text{quads}} \geq 0, u_i \in [0,1], v_i \in [0,1], \end{aligned} \tag{3.6}$$

where we dropped variable dependencies for conciseness. These terms are motivated from observations regarding the kinetics of the deployment illustrated in Figure 3.12.

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 3.14). 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 3.12

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

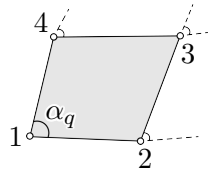
$$\mathcal{E}_{\text{len}}(\mathbf{c}, \mathbf{u}, \mathbf{v}) := \frac{w_{\text{len}}}{2\ \bar{l}_0^{\ 2}} \sum_c \sum_i \left(\|\mathbf{c}_{c,i+1} - \mathbf{c}_{c,i}\| - \|S(u_{c,i+1}, v_{c,i+1}) - S(u_{c,i}, v_{c,i})\| \right)^2, \quad (3.7)$$

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 3.9, 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 3.12). We capture this behavior with a term that minimizes this variance around each joint as

$$\mathcal{E}_\alpha(\mathbf{c}, \mathbf{u}, \mathbf{v}) := \frac{w_\alpha}{2} \mathbb{V}\text{ar}_q[\Delta\alpha_q]. \quad (3.8)$$

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



$$\Delta\alpha_q := \angle(\bar{\mathbf{c}}_{q,2} - \bar{\mathbf{c}}_{q,1}, \bar{\mathbf{c}}_{q,4} - \bar{\mathbf{c}}_{q,1}) - \angle(\mathbf{c}_{q,2} - \mathbf{c}_{q,1}, \mathbf{c}_{q,4} - \mathbf{c}_{q,1}), \quad (3.9)$$

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

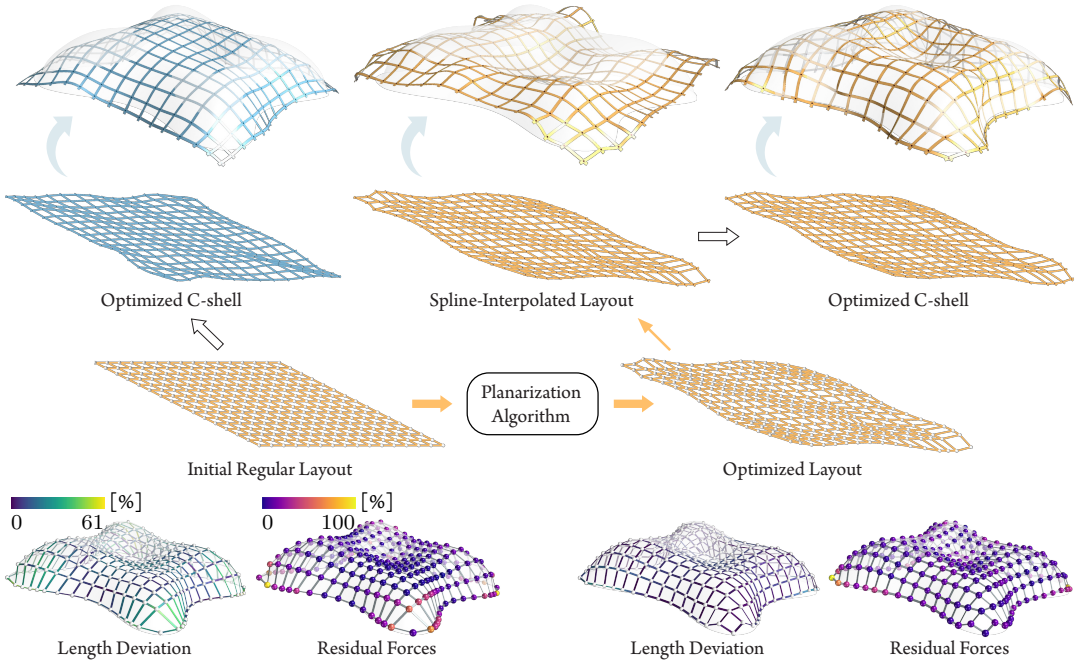


Figure 3.14: Our planarization is essential to get a good initialization for design optimization. When starting from a regular grid (left), the design optimization fails to capture the bumps of the target surface adequately. The flattening algorithm computes an initial layout that does not accurately 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.

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 force balance equation residuals are given by

$$\mathcal{E}_{\text{eq}}(\mathbf{c}, \mathbf{u}, \mathbf{v}) := \frac{w_{\text{eq}}}{2(YA_0)^2} \left\| \frac{\partial E}{\partial \bar{\mathbf{c}}} - \lambda^* \frac{\partial \bar{\alpha}}{\partial \bar{\mathbf{c}}} \right\|^2, \tag{3.10}$$

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

$$\mathcal{E}_{\text{bnd}}(\mathbf{u}, \mathbf{v}) := \frac{w_{\text{bnd}}}{2\ \bar{l}_0^{\ 2}} \sum_i \left\| S(u_{b,i}, v_{b,i}) - \bar{\mathbf{c}}_{b,i}^{(\text{tar})} \right\|^2 \tag{3.11}$$

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 3.13 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 3.14 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]).

3.7 Numerical Solver

We solve both the design optimization problem in Equation (3.3) and the planarization problem in Equation (3.6) using Sequential Linear-Quadratic Programming (SLQP), a trust-region active-set method [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 $\bar{J}(\mathbf{q}, \mathbf{p}, \bar{\alpha}^{\text{tgt}}) := J(\mathbf{x}_{\text{3D}}^*(\mathbf{p}, \bar{\alpha}^{\text{tgt}}), \mathbf{q}, \mathbf{p})$ with respect to the DER rest-state quantities $\mathbf{p}$ and deployment angle $\bar{\alpha}^{\text{tgt}}$ using adjoint sensitivity analysis as detailed in Appendix A. 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, $\frac{\partial \bar{J}(\mathbf{q}, \mathbf{p}(\mathbf{q}), \bar{\alpha}^{\text{tgt}})}{\partial \mathbf{q}} = \frac{\partial \bar{J}}{\partial \mathbf{p}} \frac{\partial \mathbf{p}}{\partial \mathbf{q}} + \frac{\partial \bar{J}}{\partial \mathbf{q}}$.

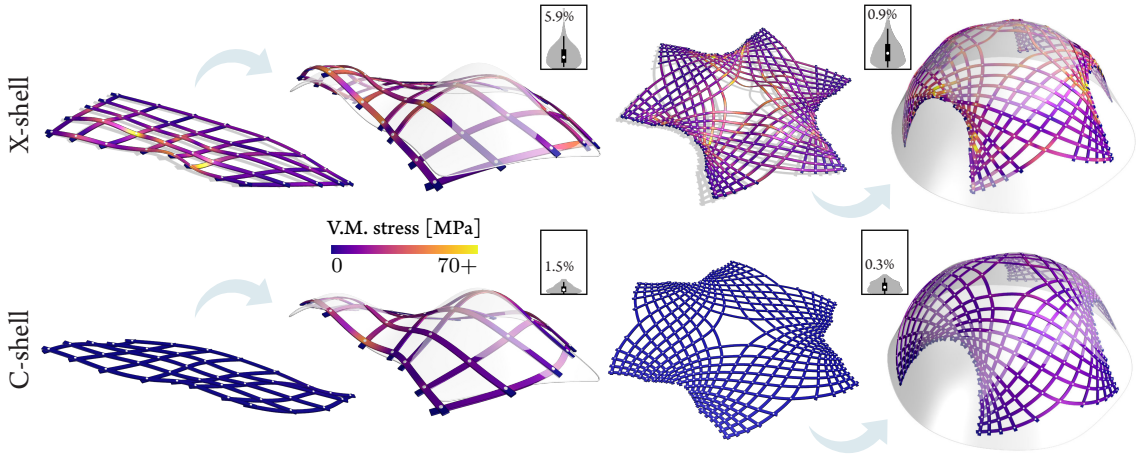


Figure 3.15: X-shells vs. C-shells. The rest quantities and target shape have been chosen such that both deployed gridshells approximate the target well. The crucial difference is that X-shells have significant stress in the rest state, which no longer remains planar. The von Mises scale is cropped to 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.

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 Appendix A. 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.

3.8 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 3.15), reduce stresses in the deployed state (Figure 3.6), and allow for more

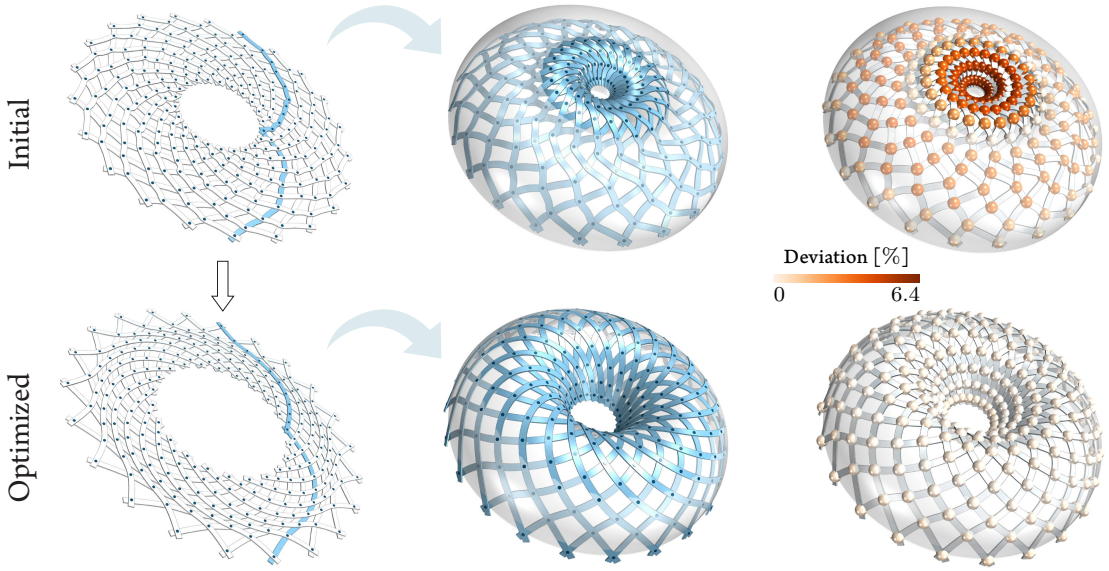


Figure 3.16: Exploiting symmetry. For a symmetric target surface such as the torus, we explicitly enforce beams to be identical in each linkage family using a reduced representation highlighted in the planar layouts. The initial design (left) is optimized to best approximate the target (right). The deviation to the target surface is indicated as a percentage of the bounding box diagonal at each joint.

flexibility in terms of linkage topology (Figures 3.3 and 3.21). 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 3.1 and 3.20). 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 3.15, this leads to high stress in the

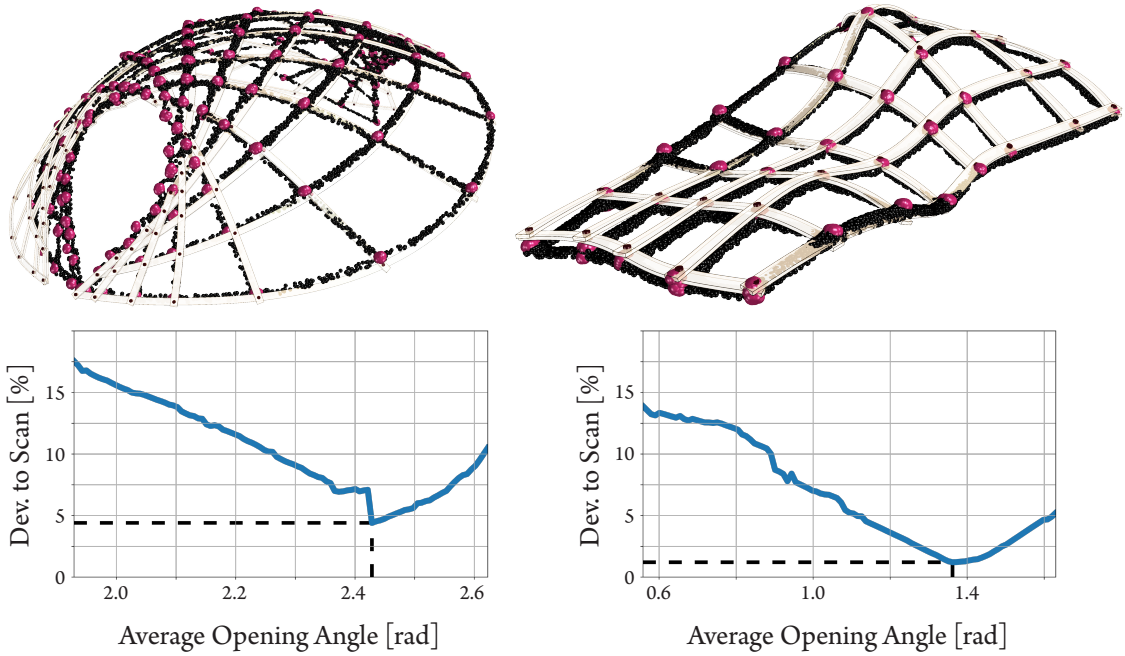


Figure 3.17: The simulated models show good agreement with the 3D scans of the fabricated prototypes. The physical models are deployed by hand, and held in place by either attaching strings to constrain the distance between pairs of joints (left), or by fastening the screws at the joints (right). Since 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.

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 3.17).

Symmetric Designs Figure 3.16 shows a design where we explicitly impose identical beam shapes in each linkage family. We implement this design constraint using a reduced represen-

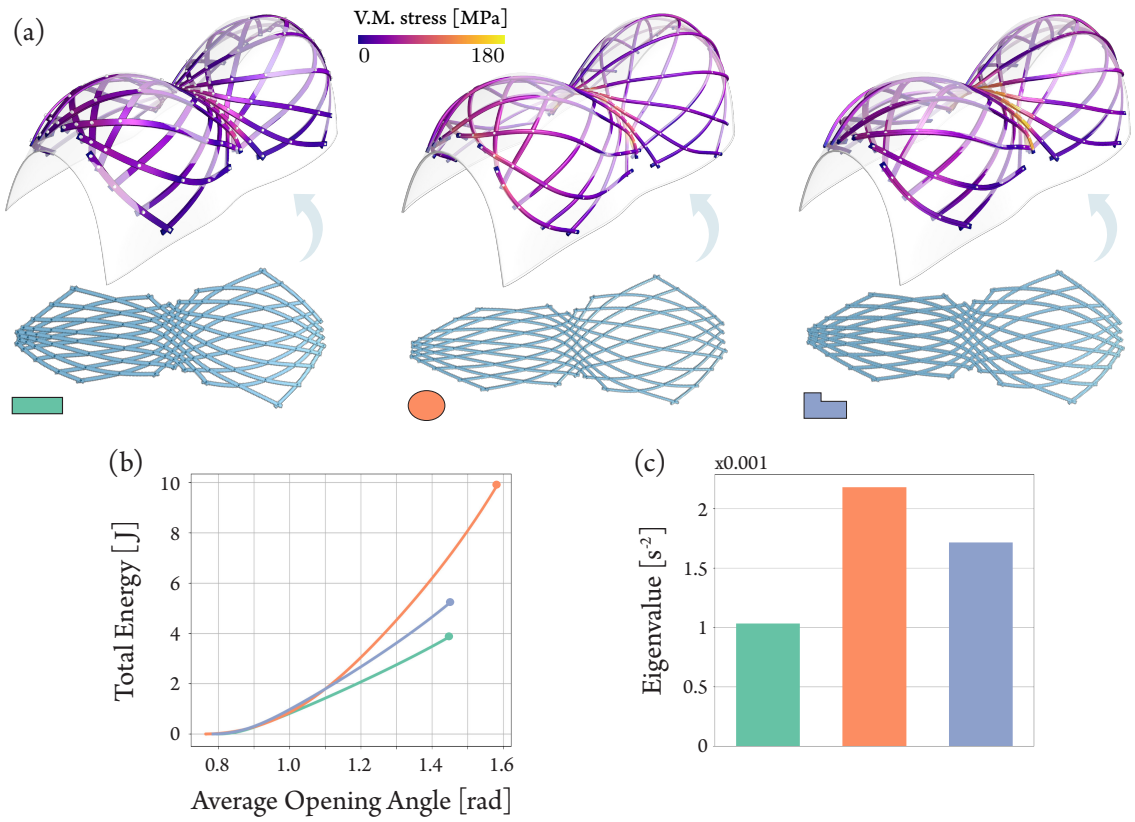


Figure 3.18: Effect of varying the cross-section. We optimize an initial linkage towards the same target surface using three different cross-sections, all sharing the same area. The resulting linkages have a comparable median deviation to the target surface: Respectively 0.16%, 0.17%, 0.15% of the bounding 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.

tation 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 3.18 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 $H\mathbf{v} = \mu M\mathbf{v}$,

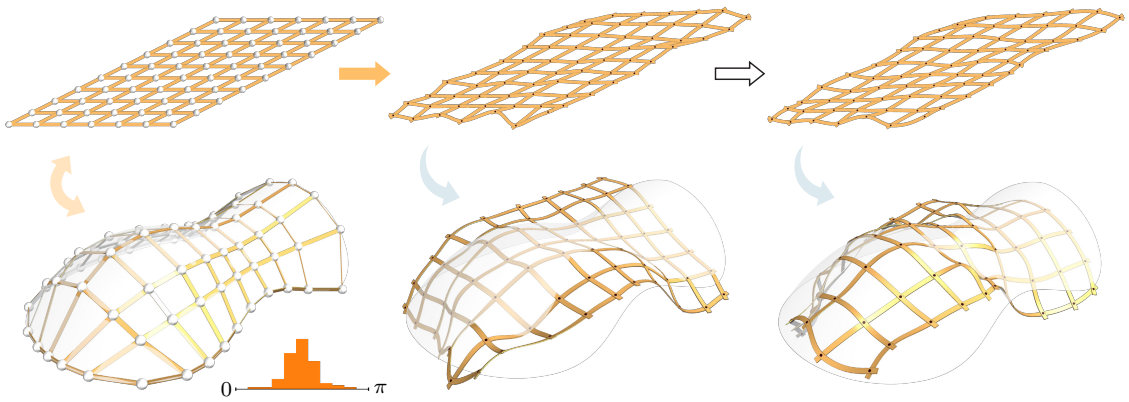


Figure 3.19: A failure case for our inverse design pipeline. Due to the smoothness of the boundary, the opening angles must span a large spectrum as shown at the top, which is incompatible with highly curved deployed states (see Figure 3.9).

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 3.1 and Table 3.2.

3.9 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, 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 We do not explicitly prevent self-intersections of beams during design optimization or deployment. Our regularization aims at reducing the risk of self-penetration, yet we cannot guarantee an intersection-free final result. Manual intervention is required if self-intersections are encountered, such as increasing ϵ and the regularization weights. While in all

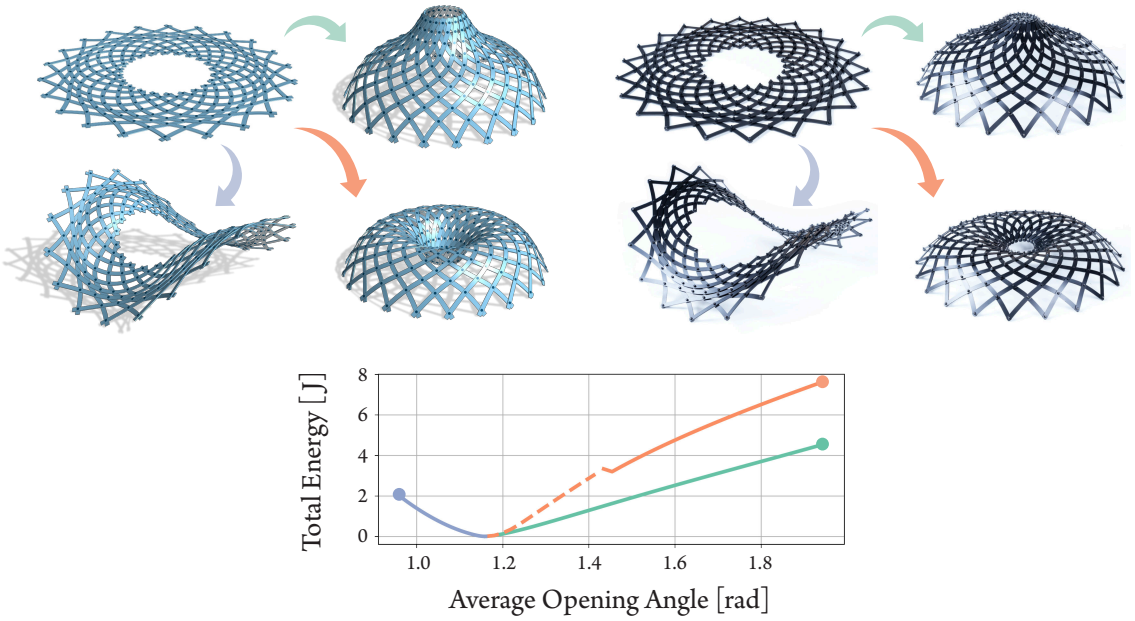


Figure 3.20: Multi-deployment. This model has been optimized towards a toroidal shape (orange), but can also deploy towards a cone-like shape (green) and a saddle-like surface (purple). The energy plot indicates that the toroidal shape has higher energy than the cone shape. Hence, additional external 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.

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.

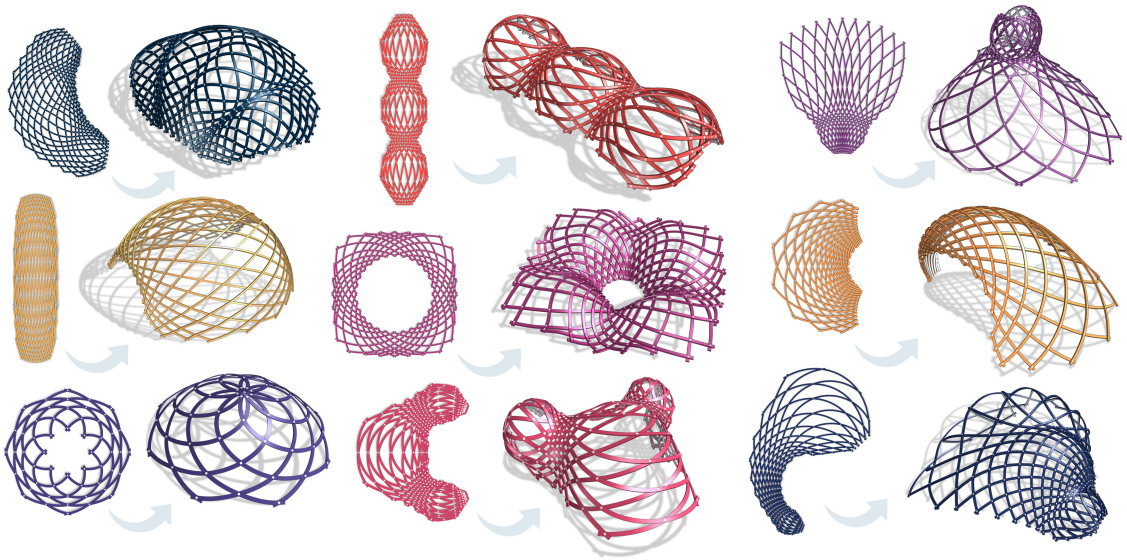


Figure 3.21: C-shells offer increased flexibility in terms of linkage topology, which leads to a significantly richer shape space compared to deployable linkages with straight beams.

Planarizeable Target Surfaces 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 3.16. As illustrated in Figure 3.19, 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 3.20, 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.

3.10 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

Table 3.1: Timings 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.

Models (Figures)	#J	#R	n_q	Sim. Time	Opt. Time
Dome (3.1, 3.11)	142	30	986	00:16	49:20
3Bumps, w_T=6×10^3 (3.5)	127	26	866	00:22	07:26
3Bumps, w_T=6×10^4 (3.5)	127	26	866	00:23	03:12
3Bumps, w_T=6×10^5 (3.5)	127	26	866	00:24	03:51
3Bumps, w_T=6×10^6 (3.5)	127	26	866	00:24	03:31
Square Boundary (3.7)	312	48	1824	00:34	1:05:42
Circular Boundary (3.7)	312	48	1824	00:34	5:25:04
Bat, w_{ip}=0 (3.8)	127	26	710	00:20	15:16
Bat, w_{ip}=1×10^3 (3.8)	127	26	710	00:14	07:26
Bat, w_{ip}=5×10^3 (3.8)	127	26	710	00:16	16:51
Hexagon (3.15)	453	66	2694	01:05	2:22:25
Torus (3.16)	242	44	62	00:42	19:40
2Bumps, rect. (3.18)	127	26	920	00:15	31:36
2Bumps, ellipsoidal (3.18)	127	26	920	00:08	33:34
2Bumps, L-shaped (3.18)	127	26	920	00:11	34:56

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.

Table 3.2: Quantitative assessment of the design optimization on several models. *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)	Tgt. Fit. (%)	Final En. (%)	Final Obj. (%)
Dome (3.1, 3.11)	16.91	109.76	46.61
3Bumps, w_T=6×10^3 (3.5)	67.68	43.22	47.05
3Bumps, w_T=6×10^4 (3.5)	48.66	65.76	51.25
3Bumps, w_T=6×10^5 (3.5)	47.58	96.16	27.84
3Bumps, w_T=6×10^6 (3.5)	41.69	118.85	16.36
Square Boundary (3.7)	19.18	88.03	15.00
Circular Boundary (3.7)	8.28	105.11	0.79
Bat, w_{ip}=0 (3.8)	104.95	43.90	65.72
Bat, w_{ip}=1×10^3 (3.8)	110.64	65.49	71.61
Bat, w_{ip}=5×10^3 (3.8)	159.22	69.18	52.65
Hexagon (3.15)	25.69	16.91	20.57
Torus (3.16)	10.71	14.20	5.26
2Bumps, rect. (3.18)	39.54	76.94	35.95
2Bumps, ellipsoidal (3.18)	44.28	97.95	32.08
2Bumps, L-shaped (3.18)	56.07	92.14	44.60

Note

This chapter is based on the following publication:

> **Quentin Becker**, Seiichi Suzuki, Yingying Ren, Davide Pellis, Julian Panetta, Mark Pauly. C-shells: Deployable Gridshells with Curved Beams. *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia 2023)*, Best Paper Award Honorable Mention.

The candidate contributed most of the scientific developments and implementation of this publication.

Chapter 4

Interactive Design of C-shells Using Reduced Parametric Families

Elastic gridshells are a class of planar-to-3D linkages typically composed of straight flexible beams connected via rigid rotational joints. The integration of planar curved flexible beams into these linkages significantly broadens their design space, allowing for a diverse range of deployable configurations. We termed these special types of linkages *C-shells*. The main characteristic of these structures is that they intrinsically encode their deployed states in the shapes of their beams while maintaining a planar and zero-energy assembly state. However, designing a C-shell is not a trivial task due to the freeform shape of their beams and the complex physical behavior emerging from their elastic deformation. To address this challenge, we introduce a computational approach for interactive design exploration of C-shells. We show how we can leverage geometric principles on the arrangement of the beams to efficiently create linkage assemblies that deploy into low-energy states of varied geometries and aesthetics. Our reduced parameterization relies on planar conformal maps that preserve the deployability of a reference linkage while generating a spectrum of smooth design variations. To explore design alternatives, users can interactively modify the beam connectivity, starting from a selection of predefined parametric grid topology families. Our interactive tool includes a physics simulation, offering both visual and quantitative feedback on the deployment of the explored designs. We showcase the utility and adaptability of our approach through design studies of a variety of C-shell families.

4.1 Introduction

Elastic gridshells are a prominent application of the principle of active bending and are characterized by thin, flexible beams interconnected at scissor joints to form a grid-like structure. This arrangement facilitates intricate deformation behaviors, with beams undergoing large elastic deformation at small strains while rotating around their joints.

Historically, gridshells have utilized straight beams arranged in uniform grids that are deformed to curved 3D geometries by imposing suitable boundary constraints. Recent advancements depart from this shaping principle and instead, arrange the beams in non-uniform configurations. These arrangements introduce kinetic incompatibilities, forcing the structures to buckle out of plane

when deployed. Here, a significant advantage lies in the capability to encode target shapes directly into the material system by fine-tuning the distances at which beams connect. However, due to such irregularities, the main disadvantage is their tendency to adopt non-planar states at rest.

In this context, C-shells emerge as a novel class of non-uniform elastic gridshells characterized by the use of curved planar beams with arbitrary cross-sections. Offering a broader design space for constructing free-form surface geometries, C-shells guarantee a planar and stress-free assembly state due to the introduction of rest curvature in the planar beams. In this paper, we introduce an approach for the design exploration of these structures using a low-dimensional representation of the beam layout. Our reduced parameterization relies on planar conformal maps, preserving the deployability of a reference linkage while generating a spectrum of smooth design variations. We showcase the utility and adaptability of our approach through various C-shell families.

4.2 Related Work

Smooth Chebyshev nets are the fundamental principle for designing regular elastic gridshells. The most common geometric technique used in architecture to address the design of these gridshells is the so-called compass method, initially proposed by Burkhart and colleagues [Burkhardt and Otto, 1978]. Alternative approaches have been proposed based on the duality of isoradial meshes and Chebyshev nets [Douthe et al., 2017], the combination of basic buildings blocks [Baek et al., 2018] and multi-patch approaches based on optimization methods to approximate intricate geometries [Garg et al., 2014, Sageman-Furnas et al., 2019]. Special attention demands the concept of C-meshes proposed by Liu et al. [Liu et al., 2023a]. C-meshes are classes of deployable grid-shell structures with beams placed orthogonally to a target surface. One particular feature is the existence of C-meshes shaped from straight strips for Chebyshev nets approximating surfaces of constant Gaussian curvature and circular strips for quad grids approximating surfaces with the linear-Weingarten property.

Relevant to our work is the research on non-regular elastic grid-shell structures. X-shells [Panetta et al., 2019] employ straight beams arranged in irregular grids, enabling assembly in a near-planar configuration before deployment into a 3D shape. This method offers designers the ability to iteratively adjust the planar grid layout based on interactive feedback regarding the resulting 3D geometry. Subsequently, optimization is applied to minimize elastic energy in both the flat assembly and deployed states. Recent research on elastic geodesic grids has similarly focused on exploring irregular grid configurations [Pillwein et al., 2020, Pillwein and Musialski, 2021, Soriano et al., 2019]. These studies take advantage of the inherent property of thin lamellas, which are

constrained to conform to geodesic curves on a surface owing to their highly anisotropic bending stiffness. Schling and colleagues [Schling et al., 2018] investigated a distinct type of non-regular grid-shell structure employing asymptotic grids to approximate surfaces with non-positive Gaussian curvature.

The utilization of curved planar ribbons is another aspect relevant to our work. Ren et al. [Ren et al., 2021] introduced a method that extends weaving techniques to incorporate curved ribbons, offering greater flexibility in the topology of the weaving pattern and resulting in a smoother appearance of the final woven structure. Baek and coworkers [Baek et al., 2021] explored the impact of in-plane curvatures of ribbons to develop smooth spherical, ellipsoidal, and toroidal triaxial woven structures using a unit-cell approach. Mhatre et al. [Mhatre et al., 2021] proposed a deployable circular structure made from elastic beams with constant in-plane curvature. In contrast to our approach, these methods do not explicitly couple beams along their span with rotational joints.

4.3 Background

We briefly describe the way C-shells are defined, simulated, and deployed as initially proposed in [Becker et al., 2023]. More details on these aspects can be found in the original paper.

C-shells Parameterization C-shells are instantiated by manipulating a set of control points in the plane that are interpolated using cubic splines [Farin, 2002], each representing a curved elastic rod. These control points can be further categorized into joints, which are shared among splines, and intermediate control points, which are specific to each spline. We gather both in a single array $\mathbf{q} \in \mathbb{R}^{n_q}$ that defines the degrees of freedom of the system. This parameterization decouples the topology of C-shells from the geometric information encoded in the beam shapes.

Physical Model Each beam is split into segments at the joints, which are then simulated using the elastic rod model from Bergou and colleagues [Bergou et al., 2008, 2010]. At each end of each segment, a rotational joint connects beams belonging to different families by enforcing positional and rotational constraints except for a rotation about the joint normals. Elastic rods are defined by a rest quantities array $\mathbf{p}(\mathbf{q}) \in \mathbb{R}^{n_p}$ (rest lengths and rest geodesic curvatures) extracted from the splines directly so that any layout is at rest and flat when drawn by the designer. The segments are further decomposed into edges, whose positions and orientations are stored in the array $\mathbf{x} \in \mathbb{R}^{n_x}$. Special treatment is given to edges connected at joints, as described in Panetta et al. [Panetta et al.,

2019]. Deformations of the linkage are quantified using the elastic energy $E(\mathbf{x}, \mathbf{p})$ that accounts for stretching, bending, and twisting of the linkage beams.

C-shells Deployment Deployment is achieved by imposing uniform torque at the joints and solving for the equilibrium state $\mathbf{x}^*_{3\mathrm{D}}$ of the whole linkage. Equivalently, we can constrain the average opening angle $\bar{\alpha}(\mathbf{x})$ to a user-defined value [Becker et al., 2023]. The deployment process is formulated as the following optimization problem

$$\mathbf{x}^*_{3\mathrm{D}}(\mathbf{p}, \bar{\alpha}^{\mathrm{tgt}}) := \underset{\mathbf{x}}{\operatorname{argmin}}\, E(\mathbf{x}, \mathbf{p}) \qquad (4.1)$$
$$\text{s.t.} \quad \bar{\alpha}(\mathbf{x}) = \bar{\alpha}^{\mathrm{tgt}},$$

where $\bar{\alpha}^{\mathrm{tgt}}$ is the target average opening angle. We additionally pin the three translational and three rotational degrees of freedom of an arbitrary node to eliminate global rigid transformations.

4.4 Geometry for C-shells

Due to the complex interplay of beam connections and the freeform geometry of beam shapes, the physical deployment of C-shells offers rich geometric behavior that can be difficult to predict by the designer. To aid interactive design, we have found that certain geometric principles based on joint angles and inter-joint distances allow us to qualify the structure's deployability and the sensitivity of its deployed state to the input parameters.

Opening Angles Distributions The deployment of a C-shell defined in Equation (4.1) is determined by its average opening angle. Therefore, the distribution of joint angles in the planar rest state naturally impacts the deployability of the structure.

First, the range of this distribution determines whether the C-shell can be deployed by increasing or decreasing joint angles. The difference between the maximum value in the range and π indicates the extent to which the angles can be opened for deployment. Similarly, the difference between the minimum value in the range and 0 represents the degree to which the angles can be closed for deployment, see Figure 3.9.

Second, the extent of the range of the distribution gives an indication of the incompatibility of a C-shell. A narrow angle spectrum indicates a compatible linkage (which stays in plane when deployed) or a conformal variation of one such linkage. On the other hand, a wider opening angle spectrum indicates an incompatible linkage in the vast majority of the cases. For instance,

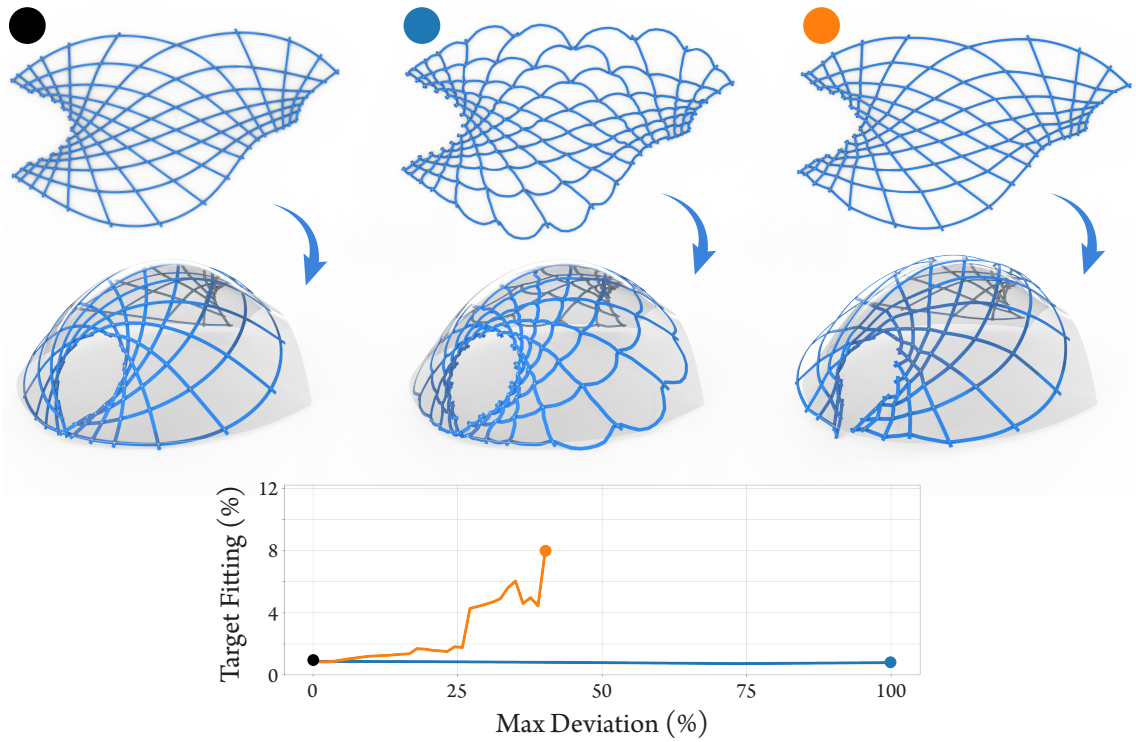


Figure 4.1: The curvature of segments at rest does not significantly impact the deployed shape if the joint's positions are identical. Designs on the left and on the center share the same joints with varying intermediate control points. The closest deployed state along the deployment path (second row) remains close to the target surface. However, applying a random perturbation to each joint position independently (rightmost design) leads to a significant deviation from the original surface. We report the target deviation as a percentage of the bounding box diagonal of the target surface as a function of the perturbation magnitude.

a Klann-like linkage [Klann, 2002] exhibits varying opening angles while remaining compatible. However, any random perturbations to the initial design will likely compromise this property. In practice, they represent a negligible fraction of the designs, which makes the distribution width a valid measure of incompatibility in a linkage.

Inter-Joint Distances C-shells are parameterized using splines that interpolate joint positions. While intermediate control points are introduced to enrich the space of beam designs, we show in Figure 4.1 that the deployed shape is less sensitive to their positioning. Furthermore, we observe that a relatively small uncoordinated perturbation of the joints' positions can lead to a significant deviation from the target surface. The overall form of a C-shell is predominantly determined by the arrangement of joints in the plane, rather than the specific placement of intermediate control

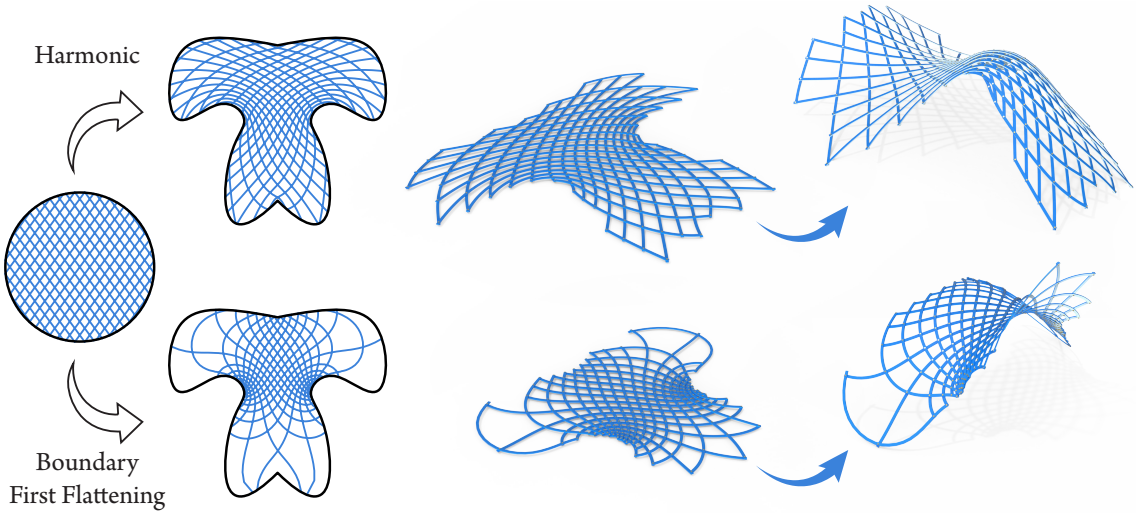


Figure 4.2: Our tool allows the user to manipulate the domain boundary and deform an initial grid layout according to two strategies. Harmonic parameterization [Eck et al., 1995] is near-conformal while Boundary First Flattening [Sawhney and Crane, 2017] deforms the input conformally. Larger quadrilaterals are filtered out depending on a user-defined tolerance on their areas. The resulting models can be deployed for further inspection.

points.

4.5 Parametric Families of Deformations

Based on the foundational concepts of C-shells parameterization and deployment, we introduce a practical approach for exploratory design processes. Our method involves a series of interactive editing operations that enable the deployability-preserving and smooth deformation of an initial 2D linkage design. These 2D deformation maps can be combined arbitrarily to generate design variations. Our physics-based linkage simulation then allows evaluation of the design modifications on the deployed 3D structure.

Conformal Deformations

Conformal maps preserve the angle distribution and distort the inter-joint distances of an input C-shell layout. In light of the considerations above, they hence allow controlling the deployed shape while preserving the deployability of the initial design.

Planar conformal maps offer a rich space of 2D deformations. The Riemann mapping theorem states that given two simply connected subsets of $\mathbb{R}^2$, there exists a bijective conformal map between them [Walsh, 1973]. Such a mapping is unique up to rotations and translations. To control

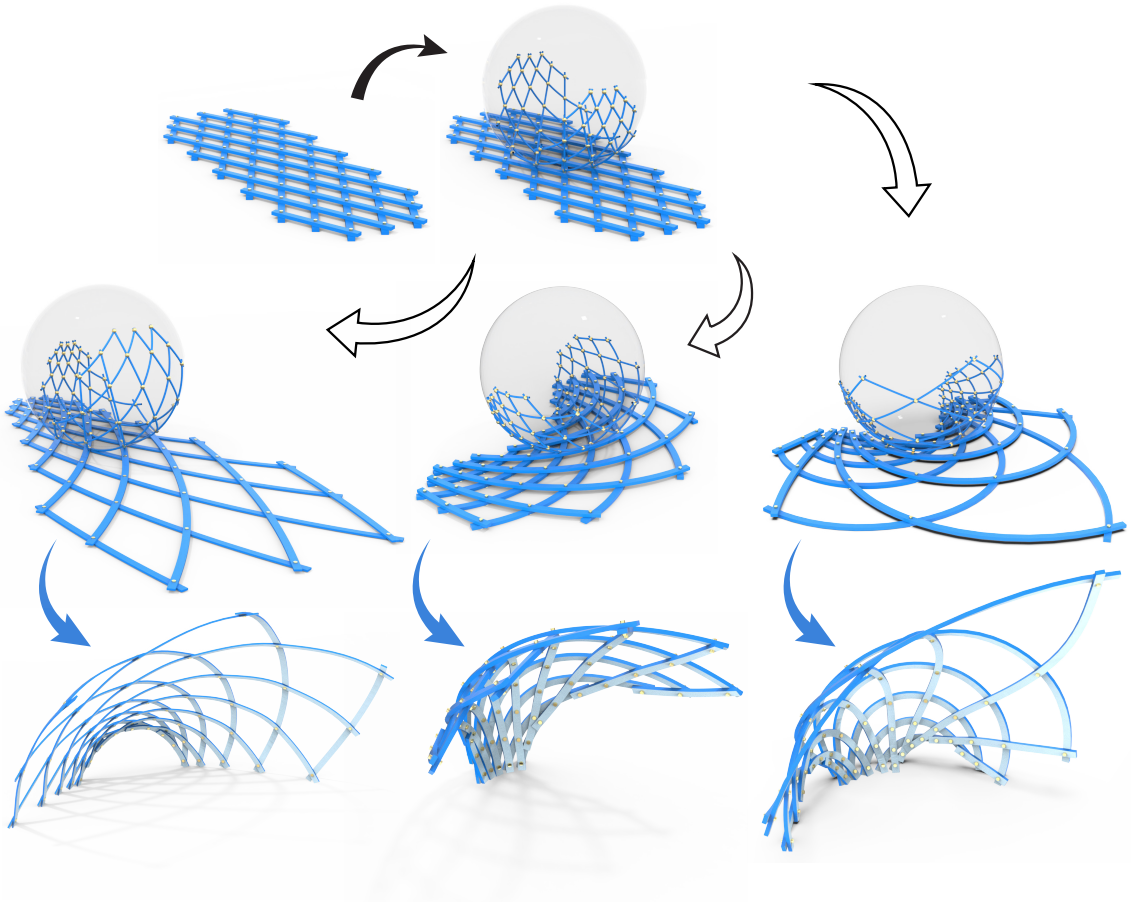


Figure 4.3: Three variations (bottom two rows) of a reference linkage template (top) using 2D Möbius transformations. The initial layout is projected onto the sphere, rotated about the sphere center, and projected back to the plane.

such a conformal map, the user can manipulate the non-self-intersecting closed boundary curve of the linkage. We can then infer the corresponding conformal map using the Boundary First Flattening (BFF) algorithm [Sawhney and Crane, 2017] applied to planar domains. This method solves for a discrete approximation of a conformal map that we apply to our linkage structure to obtain the new beam geometries and joint locations.

Alternatively, we can replace the conformal map with a harmonic parameterization [Eck et al., 1995]. While harmonic maps do not preserve angles, they can reduce area distortion, which leads to a more uniform spacing of joints. Both methods are shown in Figure 4.2, where we illustrate how a simple regular grid on a unit disk can be transformed to create interesting design variations.

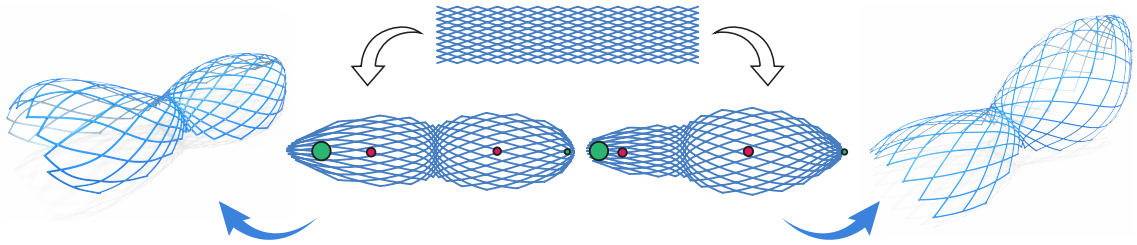


Figure 4.4: A reference linkage (top) is deformed using attractive and repulsive points laid out in the plane with varying action strengths. Green points attract neighboring joints with an intensity proportional to their radii, similar to red points repulsing neighboring joints. Gaussian RBFs with a single shared extent and varying strengths are chosen in these designs.

Möbius Transforms As an alternative to editing the linkage boundary, we offer a more direct way to define a specific subset of conformal maps, called *Möbius transforms*. These maps can be created as a composition of three basic transforms: An inverse stereographic projection onto a sphere, a rotation of the sphere about its center, and a projection back to the plane. We expose three parameters to the user: The height of the sphere center and the rotations about the initial plane axes. The first parameter controls the scale of the initial linkage, while the last two parameters allow bringing various kinds of incompatibilities into the linkage. We conceal the x and y coordinates of the sphere center and the rotation about the normal to the design plane since they only rigidly deform the initial layout. Figure 4.3 illustrates how this simple interface enables effective exploration of design alternatives.

Attractive and Repulsive Points

Conformal deformations primarily manipulate the geometric properties at a global level, while preserving the deployability of the grid layout. We propose an alternative strategy that allows direct manipulation of joint positions using local attraction or repulsion forces defined on a small set of local control points. Deformation fields generated from such point sources are determined by the choice of a Radial Basis Function (RBF). The Gaussian RBFs used in Figure 4.4 allow decoupling interaction strength s and interaction extent σ for each point on a joint located at a distance r as $\frac{s}{\sigma\sqrt{2\pi}} \exp\left(-\frac{r^2}{2\sigma}\right)$. A positive s indicates a radially repulsive point, while a negative value defines a radially attractive point. The strength and extent of the points actions can be adjusted dynamically, allowing for real-time user interaction. This addition enhances the versatility and applicability of the proposed approach, offering designers a comprehensive toolkit for exploring C-shell layouts.

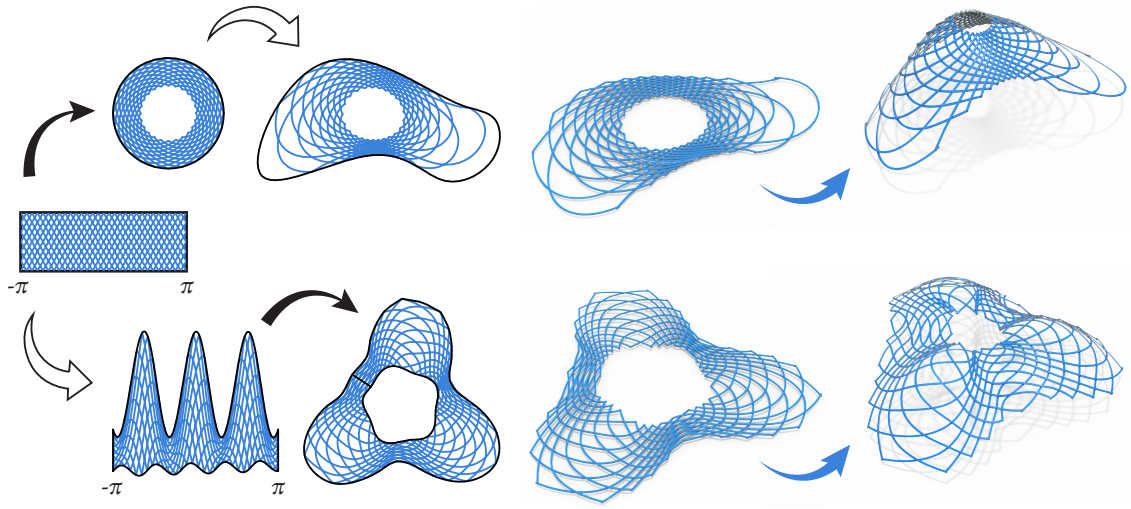


Figure 4.5: Our boundary-based deformations are compatible with other linkage representation spaces, allowing for finer control on annulus topologies. Converting a rectangular lattice linkage topology made of rhombus cells in the polar coordinates domain (left) to the Euclidean space produces a linkage with an annulus topology (top row, left). Boundary First Flattening [Sawhney and Crane, 2017] only allows simply connected domains to be manipulated and discards the inner boundary. The designer can edit the outer ring exclusively. Instead, we let the user edit the rectangular boundary in the polar coordinates domain before pushing the design to the Euclidean space (bottom row). This allows the inner and outer rings to be manipulated independently.

4.6 Linkage Topology Design

In the preceding section, we introduced techniques for manipulating joint angles and distances, offering means to adjust the geometry of the planar linkage layouts. Expanding on these methods, we now turn our attention to the topology of linkage grids. Navigating the full space of linkage topologies is difficult due to the combinatorial explosion of possibilities. We therefore expose a subset of such linkage topologies and facilitate direct user edits supported by automatic deletion of joints based on geometric criteria and tolerances.

Topology Families

Rectangular Lattice Topology One straightforward approach is to use either rectangular or rhombic unit cells and tile them following a rectangular lattice in a 2D Euclidean space, see Figure 4.6. The number of joints and the spacing in the x and y directions can be tuned independently before applying our in-plane deformations tools discussed above.

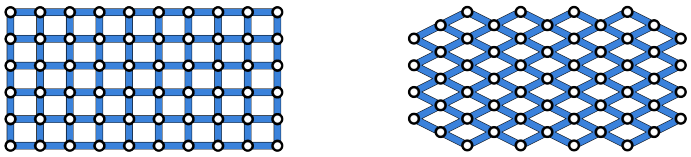


Figure 4.6: Rectangular lattices with rectangular (left) and rhombic (right) unit cells.

Annulus Topology In addition to this, we allow users to design radially symmetric designs by adapting the rectangular lattice topology to polar coordinates. Joints in the 2D space are parameterized using polar coordinates i.e., they belong to $\mathcal{S}^1 \times \mathbb{R}_+$ where $\mathcal{S}^1$ is the unit circle.

An annulus topology is initialized from a rectangular lattice topology, for which the rectangle is axis-aligned in $\mathcal{S}^1 \times \mathbb{R}_+$, with a side length of 2π along the azimuthal direction to ensure periodicity, see Figure 4.5 on the left. Overlapping joints on the rectangle sides are identified so that all curves share the same number of joints. Joint positions can be recovered in the 2D plane by evaluating the polar coordinates.

Such linkages can be edited by manipulating the domain boundary in the 2D plane or directly on the cylinder $\mathcal{S}^1 \times \mathbb{R}_+$. For the latter, we allow the designer to manipulate the sides of the rectangle along the radial direction and impose both sides to match along the azimuthal direction. This last property ensures that the deformation is well-defined on the unit circle.

While this second option affects the deployability of the linkage i.e., its opening angle distribution, it allows for a finer control of the inner circle of topology by decoupling it from the outer ring, see Figure 4.5.

Topology Editing

As the initial linkage deforms in the design process, some rod segments may grow unrealistically long. We detect such outliers and let the user discard them based on an adjustable length threshold. The remaining single valence joints are discarded iteratively until none can be found in the linkage. Alternatively, we let users manually remove joints and subsequently filter out single valence joints.

4.7 Implementation

Our C-shell design tools are packaged in a single Rhino-Grasshopper [McNeel et al., 2009] plugin, see Figure 4.7. We split them into categories: Geometric Edits, Deployment, and Design Evaluation.

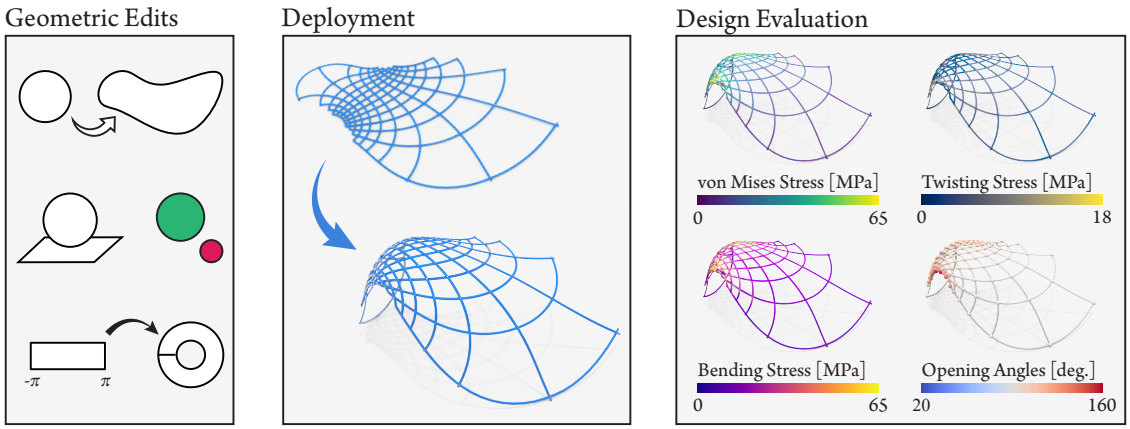


Figure 4.7: Overview of the functionalities available in our Rhino-Grasshopper [McNeel et al., 2009] plugin.

Geometric Edits encompass the abovementioned techniques, including boundary-induced deformations, Möbius transforms, attractive and repulsive points, and polar coordinates representation. The resulting linkage is simulated in the flat state and deployed according to Equation (4.1). The user controls the deployment extent, and pinned joints if any. Physics quantities can be measured and are made available to the user following the model presented in Becker et al. [Becker et al., 2023]. As an example, Figure 4.7 shows the maximum von Mises stress at each vertex of the linkage discretization, where the scale is cropped at the yield stress of Acetal Copolymer (POM-C).

4.8 Conclusion

The incorporation of planar curved beams in C-shells represents a promising alternative for the design of elastic gridshells. The research presented in this paper aims to advance the forward design process of C-shells by introducing a comprehensive framework of human-interpretable manipulations of a reference linkage, inspired by considerations of scissor linkage deployability.

Our approach enables designers to navigate the intricate landscape of C-shell design with greater ease. By offering a set of intuitive strategies based on both conformal and non-conformal deformations within various topology layouts, we facilitate real-time recombination and modification of planar linkages. Through visual feedback of the resulting deployed state, designers can intuitively assess the impact of their design decisions and iteratively refine their creations.

While our research has demonstrated the vast design space that C-shells could offer, efforts are

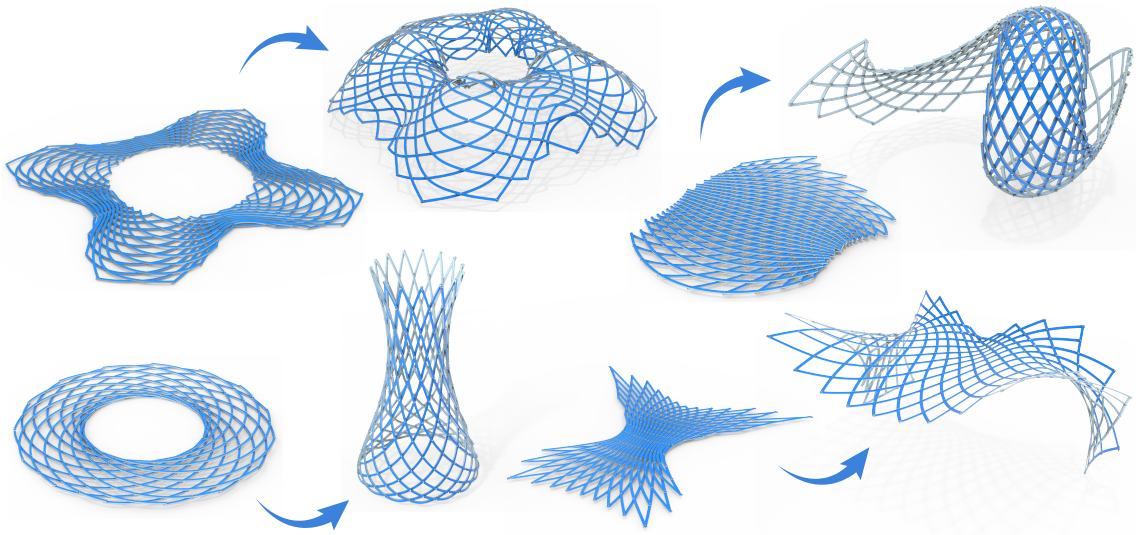


Figure 4.8: Compositions of edits produce designs with diverse-looking deployed states.

still required to address their scalability, particularly in terms of material efficiency and fabrication feasibility. However, by embracing the possibilities offered by C-shells, our research sets the stage for further innovation and exploration in the field of elastic gridshells.

Note

This chapter is based on the following publication:

> **Quentin Becker**, Seiichi Suzuki, Mark Pauly. Interactive Design of C-shells Using Reduced Parametric Families. *International Association for Shell and Spatial Structures (IASS) (Proc. of the IASS Annual Symposium 2024)*.

The candidate contributed most of the scientific developments and implementation of this publication.

Chapter 5

Computational Design of a Kit of Parts for Bending-Active Structures

Bending-active structures are composed of elastic elements that deform to achieve a desired target shape. To support effective design, inverse algorithms have been proposed that optimize the geometry of each element specifically for each design. This makes it difficult to reuse elements across designs or gain efficiency in fabrication through mass production.

We address this issue and propose a computational framework to rationalize bending-active structures into a sparse kit of parts. Our method solves for the optimal part geometry such that multiple input designs can be faithfully realized with the same kit of parts. Assigning parts to different assemblies leads to a combinatorial explosion that makes exhaustive search intractable. Instead, we propose a relaxed continuous optimization incorporating a physics-based simulation in its inner loop to model the elastic deformation of the bending-active structure accurately. Our algorithm allows analyzing different design trade-offs of a kit of parts to tune the balance between fabrication complexity and fidelity to the original designs. We demonstrate our method on three different classes of bending-active structures, showcasing the effectiveness of our approach for part reuse and sustainable practices in fabrication-driven design.

5.1 Introduction

Bending-active structures are physical systems characterized by distinctive curved geometries, which arise from the elastic deformation of initially straight or planar elements. This formation approach not only enables the creation of static structures but also facilitates the construction of kinetic and deployable systems by leveraging the reversibility of elastic deformations [Lienhard, 2014]. While the constituent elements must be thin and slender to allow significant deformation, the structures must counterbalance this material reduction to withstand loads by employing alternative stress-stiffening effects achieved through appropriate geometric design [La Magna, 2017].

The equilibrium form of these structures emerges when all internal forces induced by elastic deformation of the elements and external forces, such as gravity, are in global balance. The presence of large deformations, along with the sensitivity of the structural form to even minor changes in

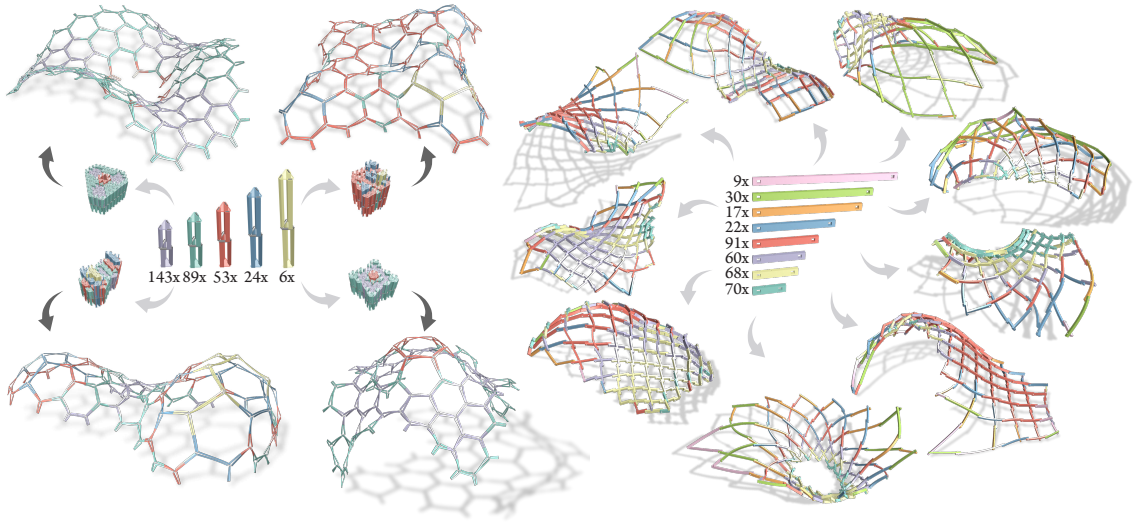


Figure 5.1: A kit of parts allows cost-effective fabrication of bending-active assemblies. Our general optimization method finds the part geometries to best reproduce a given set of input designs and can be specialized to different material systems. Left: Umbrella meshes deploy from a compact assembly state towards a target equilibrium. Right: Bending-active orthogonal gridshells assembled from straight beams that deform to best approximate the target surface.

the geometry and material properties of its constituents, makes the design process highly challenging. In response to this, physics-based simulations have been integrated into the research of bending-active structures to accurately predict equilibrium states [Lienhard et al., 2013, Manolas et al., 2022]. To match a simulated form with a desired input freeform geometry, inverse design algorithms have been proposed that directly solve for the design parameters of the bending-active structure [Pillwein and Musialski, 2021, Ren et al., 2022, Becker et al., 2023, Panetta et al., 2019]. These parameters typically define the undeformed rest state of the constituent elements, which can be fabricated in a flat state and then assembled into the final structure.

One key advantage of bending-active structures is that their constituent elements are designed to maintain strain within the elastic limits of the material, allowing for reversible elastic deformation. This means that a structure can be undeployed and individual components could in principle be re-used in a different design. However, current inverse design methods compute optimized component geometries that are specific to one particular design only. This limits the re-use potential of parts and requires custom fabrication of each individual element, which can be slow and expensive compared to mass production techniques. Our work addresses these drawbacks and investigates the question of how to design a kit of parts that can be manufactured at scale and be

re-used across multiple designs of bending-active structures.

This problem has been extensively studied for static structures composed of rigid components, in particular in the context of architectural research [Brütting et al., 2021, Alegria Mira et al., 2016]. Freeform designs are rendered feasible for fabrication by *rationalizing* them to groups of identical components. Such an optimized kit of parts can be used to assemble complex structures with an efficient fabrication pipeline, providing a cheaper and more sustainable alternative to custom fabrication. We study this problem for bending-active structures, where parts can deform into many different configurations in different assemblies. This additional complexity requires a fundamentally different approach.

Contributions.

Our main contribution is a computational framework for optimizing a kit of parts for bending-active structures. We propose a numerical method that relaxes the discrete combinatorial nature of the part-to-element assignment problem into a continuous optimization problem. This fully differentiable optimization can be seamlessly combined with a physics-based simulation that tracks the equilibrium states of all input design instances mapped onto the kit of parts. Our formulation is general in that it can be applied to different classes of bending-active structures. We show how to customize the algorithm for three concrete examples of bending-active structures, highlighting the versatility of our approach. The full source code and experiments can be found at https://go.epfl.ch/kop.

5.2 Related Work

We discuss prior work on bending-active and deployable structures in the context of component reuse and rationalization. Works proposing component reuse in architecture are followed by a review of reconfigurable systems with reusable components as their building blocks. We mention computational methods for rationalization in the context of computer graphics and further narrow our focus on modular systems involving a kit-of-parts approach to conclude the section.

Deployable Structures Deployable structures transform from a compact *rest* state that is typically easy to assemble, transport, and store to a *deployed* target state. While deployment mechanisms span across different scales and material systems [Yang et al., 2023], we focus on structures composed of elastic beams coupled via specific joining mechanisms. Trusses, space frames, and gridshells [Dyvik et al., 2021] are notable examples of such structures on an architectural scale.

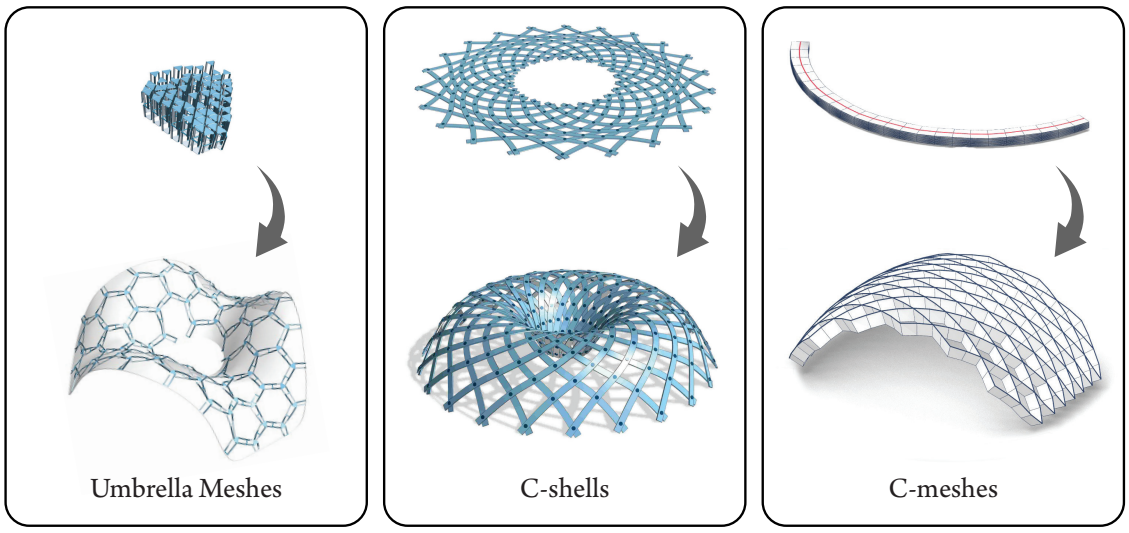


Figure 5.2: Examples of bending-active structures using elastic beams (Umbrella Meshes [Ren et al., 2022], C-shells [Becker et al., 2023]) and orthogonal grids (a superset of C-meshes [Liu et al., 2023a]) that can be deployed into a 3D target surface.

Several works deal with finding deployable variants of gridshells [Panetta et al., 2019, Becker et al., 2024, Pillwein and Musialski, 2021, Schling et al., 2022, Tellier, 2022] and the actuation sequence for their erection. The inverse-design problem of computing the rest state that deploys to a desired deployed state is of particular interest. Approaches involving geometric abstractions and numerical optimization have been proposed to solve the problem for different deployable structures [Baek et al., 2018, Ren et al., 2022, Becker et al., 2023, Liu et al., 2023a].

Figure 5.2 shows examples of bending-active deployable structures. While digital fabrication techniques have enabled the creation of bespoke solutions tailored to specific deployment states, our focus shifts to optimizing kit of parts that can be reused across multiple designs and deployments. This shift to generalized kits introduces significant challenges. Specifically, the inherent difficulty of rationalizing a material system is further intensified by the large deformations characteristic of bending-active structures. In these cases, the complexity is twofold: Ensuring that the structure's deployability is preserved throughout the rationalization process, and accurately maintaining and tracking the equilibrium states of the system.

Orthogonal Gridshells Orthogonal gridshells define another subclass of bending-active structures that is of particular interest. The beam profiles are oriented normal to the design surface, allowing beams to deform along their weak axis to approximate the surface geometry, while load

transfer occurs locally via their strong axis [Schling, 2018]. Networks of asymptotic curves to create asymptotic gridshells [Schling et al., 2022, Schling and Barthel, 2020] and pseudo-geodesic gridshells [Mesnil et al., 2023] are examples of such structures using straight strips. Circular strips have also been used on surfaces of constant mean curvature [Schling et al., 2018], or for deployable orthogonal structures (along with straight ones) through the concept of C-meshes [Liu et al., 2023a]. We consider a generalization of these structures with curved planar lamellas.

Re-usable Structures Our focus on a kit-of-parts approach addresses the challenge of promoting the reusability of structural components. Reuse-driven design [Fivet and Brütting, 2020, Huang et al., 2021] is crucial for promoting component reuse both upstream (by procuring components for future use) and downstream (by designing with future reuse in mind). An interesting approach of re-usability are modular structures made of identical universal components that can be reconfigured for various design realizations. Alegria and co-workers 2016 introduce a universal scissor component that can be reconfigured to all basic scissor cells. Liu et al. [2022] use reconfigurable units with three multi-stable states (long, short, and bent) to design 3D space metawires for reconfigurable antennas. Kusupati and colleagues 2023 use identical, shape-agnostic, and reconfigurable umbrella cells to realize structures that deploy into a large range of desired geometries. Although universal reconfigurable components can be manufactured at scale and reused across various designs, they often involve significant complexity in terms of part geometry and fabrication. In contrast, our work adopts the principle that a simpler, less reconfigurable kit of parts can be mass-produced more cost-effectively, striking a balance between part reuse and fabrication complexity.

Computational Rationalization

An important aspect of our work involves the rationalization of target geometries into a finite set of parts. Research in computer graphics and computational geometry is relevant for the panelization of free-form surfaces [Singh and Schaefer, 2010, Liu et al., 2021, Zhu et al., 2023] with applications in architecture [Eigensatz et al., 2010a, Zimmer et al., 2012]. Fu and colleagues 2010 generate a set of K quads whose instances can produce a tiled quad surface that approximates the input surface. Freeform honeycomb structures [Jiang et al., 2014] provide a torsion-free support structure with identical nodes. Jiang et al. [2021] use panels that are manufacturable by precise isometric bending of surfaces made from a few molds of constant Gaussian curvature. Various works such as [Testuz et al., 2013, Luo et al., 2015, Zhang and Balkcom, 2016] also explore volumetric rationalization of 3D shapes using shape filling blocks.

Other works focus on clustering the structural components into a set of parts based on different metrics. Basso and colleagues 2009 perform an optimization on free-form gridshells to cluster elements into a predefined number of different length groups. Liu et al. [2023b] present a clustering–optimization framework to reduce the number of different nodes in space frame structures. Zimmer and co-workers 2014 rationalize free-form shapes to a single kit of parts using the *Zometool* set composed of linear elements of nine different lengths connected by one universal joint with different connection directions. Lu and Xie [2023] reduce the number of different members in a truss layout by considering shared lengths between members as well as shared cross-sections. Schling and Barthel [2020] provide a holistic theory of *repetitive structures* considering both the geometric and constructive parameters through computational design. Their systematic study aims to identify principle relationships of form and structure and develop new design strategies.

Kit-of-Parts Approach

It is not a new idea to use a kit of parts pre-designed and engineered to be mass-produced for construction. Howe et al. [1999] draw parallels to an object-oriented programming environment with well-defined interfaces to be followed (e.g. load transfer rules, cost constraints, boundary constraints). Brütting and colleagues 2021 present a new computational workflow to design a bespoke kit of parts that can be employed to build structures of diverse typologies using optimization of structural members and joints i.e., the kit of parts that fit multiple geometric and structural requirements. St-Hilaire and Nejur [2022] propose form-matching of a temporary architectural structure with a kit of parts coupling wood with simple bendable steel strips. Gaudreault and Nejur [2023] introduce a constructive system aimed at maximizing the integration of reclaimed materials for the construction of triangular reticular structures. While these works take a kit-of-parts design approach, they do not handle free-form bending-active structures. We provide a general framework for rationalizing bending-active structures employing physics-based simulation for form-finding in the inner loop of the optimization.

5.3 Overview

Bending-active structures based on elastically deforming beams share many commonalities, even when their deployment mechanism are fundamentally different. This observation motivates our formulation of a general kit-of-parts optimization approach that can be customized towards specific classes of bending-active structures.

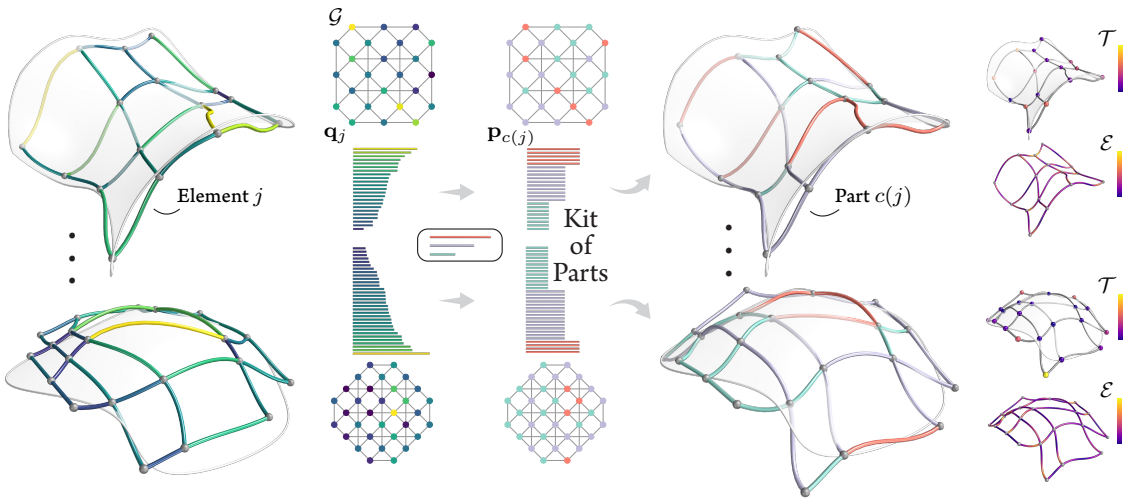


Figure 5.3: Rationalization of a bending-active structure using a kit of parts: The graph $\mathcal{G}$ defines the assembly connectivity of the design with nodes representing constituent elements/beams. Projecting these designs onto a kit of parts replaces each element j with parameters $\mathbf{q}_j$ by a part $c(j)$ from the part set with parameters $\mathbf{p}_{c(j)}$. Here $\mathbf{c}$ represents the assignment function from elements to parts. The part set and the subsequent kit of parts are computed through an optimization that minimizes an objective comprising target fitting $\mathcal{T}$ and elastic energy $\mathcal{E}$ of the equilibrium state.

Our algorithm takes as input a set of existing design instances, given by the individual geometries of all elastic elements in their rest state, and corresponding assembly graphs that define the connectivity of elements in each final structure. The goal is then to optimize for a sparse kit of parts, that is, to find the optimal geometry of each part as well as an assignment function that determines which element in each input design will be realized by which part. Such a kit of parts will be effective, if the number of parts is significantly smaller than the number of elements, while at the same time enabling faithful reproduction of the input designs.

We first define a template optimization problem in Section 5.4. Our formulation abstracts from class-specific implementation details and focuses on the core objectives that are common across different classes of bending-active structures. Specifically, we show in Section 5.5 how the combinatorial problem of assigning parts to elements can be solved with a continuous relaxation that allows integrating a physics-based simulation to track equilibrium states of the given input designs. We then illustrate in Section 5.6 how this template optimization can be supplemented with specific objectives for three bending-active structures: (i) Umbrella meshes, (ii) orthogonal grids, and (iii) C-shells. Implementation aspects of the numerical optimization are discussed in Sec-

tion 5.7 with more details provided in the supplemental material.

We show in Section 5.8 how our algorithm enables users to analyze different design choices for the optimization of a kit of parts. This helps to find the most appropriate trade-off between the complexity of the kit of parts and the deviation to the input designs.

5.4 Problem Statement

In this section we introduce terminology and formulate the general problem of optimizing a kit of parts for bending-active structures.

We assume as input a set $S_1, S_2, \ldots$ of bending-active structures that represent the variability in designs that should be realizable by the kit of parts. Each structure S_k is represented by a graph $\mathcal{G}$ whose nodes denote the elastic elements of the structure that are joined according to the connectivity defined by the graph edges. Each node has attributes $\mathbf{q}_j \in \mathbb{R}^d$, a set of continuous parameters that define the element geometry. For example, $\mathbf{q}_j$ could denote the length and width of a straight beam and the location of rotational joints along the beam. For ease of notation, we accumulate all element parameters in a vector $\mathbf{q} = (\mathbf{q}_1, \ldots, \mathbf{q}_n)$ where n is the total number of elements across all input designs.

Equilibrium Computation

To simulate the equilibrium of a structure S_k, we convert the corresponding element parameters into a discrete representation suitable for simulation. In our case, we model elastic beams using the discrete elastic rod model introduced by Bergou and colleagues 2008, 2010. Each beam is sampled with a polyline. The rest variables of the structure are then the lengths and angles of these polylines that we collect in a vector $\mathbf{r}$. The simulation variables representing a deformed state of the design, i.e., the nodal positions and local frames of all discrete elastic rods, are collected in a vector $\mathbf{x}$. The elastic energy of the deformation is defined as $\mathcal{E}(\mathbf{x}, \mathbf{r})$ and combines stretching, bending, and twisting terms as proposed in [Bergou et al., 2010].

The deformed state of S_k at equilibrium is the solution $\mathbf{x}^*$ of a constrained minimization problem. The optimization objective combines the elastic deformation energy $\mathcal{E}(\mathbf{x}, \mathbf{r})$ with external deployment forces modeled by an energy term $\mathcal{D}(\mathbf{x})$. We also integrate Dirichlet constraints to fix a certain subset of the deformed state variables $\mathbf{x}_f \subset \mathbf{x}$ to user-specified target values $\mathbf{x}_f^{\mathrm{tgt}}$, allowing pinning vertices to fixed positions or simulating deployment. We aggregate the rest variables with the fixed variables in a vector of design variables $\mathbf{d}$. The equilibrium state $\mathbf{x}_{\mathrm{3D}}^*$ is defined

as a function of design variables $\mathbf{d}$ as

$$\mathbf{x}^*(\mathbf{d}) := \underset{\mathbf{x}}{\operatorname{argmin}}\, \mathcal{E}(\mathbf{x}, \mathbf{r}) + \mathcal{D}(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{x}_f = \mathbf{x}_f^{\text{tgt}}. \tag{5.1}$$

Kit of Parts Objective

A kit of parts is an ensemble of m parts $\mathbf{p} := (\mathbf{p}_1, \ldots, \mathbf{p}_m)$, where $\mathbf{p}_i \in \mathbb{R}^d$ define the parts' geometry analogous, but potentially different to the parameterization used for elements of the input designs.

To realize the structures S_k with the kit of parts $\mathbf{p}$, we define an assignment function $c : [1, n] \to [1, m]$ that indicates which part of $\mathbf{p}$ is assigned to which element in $\mathbf{q}$. These assignments are aggregated over all structures in a vector $\mathbf{c} = (c(1), \ldots, c(n))$. The assignment process is described in more detail in Section 5.5. Figure 5.3 illustrates $\mathbf{q}$, $\mathbf{p}$, and $\mathbf{c}$ for a simple bending-active system. See also Figure 5.6 and Section 5.6 for the specific classes we consider below. p

The optimization aims to compute the part parameters of $\mathbf{p}$ with $m \ll n$ and the corresponding assignment $\mathbf{c}$ such that element $\mathbf{q}_j$ can be rationalized as an instance of part $\mathbf{p}_{c(j)}$. This projection onto the kit of parts inevitably incurs a deviation in the resulting equilibrium shapes from the input designs. Our goal is to reduce this discrepancy to a minimum while retaining a low elastic energy of the system. We thus formulate a design preservation energy as a function of $(\mathbf{x}, \mathbf{r})$ as,

$$\mathcal{F}(\mathbf{x}, \mathbf{r}) = \mathcal{T}(\mathbf{x}) + \mathcal{E}(\mathbf{x}, \mathbf{r}), \tag{5.2}$$

where $\mathcal{T}$ is a target fitting term measuring the distance of the deformed state $\mathbf{x}$ to a given target surface, and $\mathcal{E}$ is the elastic energy of $\mathbf{x}$. In Appendix B, we describe how to apply suitable weight factors for these terms to make $\mathcal{F}$ scale-invariant. We discuss how $\mathcal{F}$ can be adapted to different systems in Section 5.6.

Once a design S_k is rationalized using the kit of parts $\mathbf{p}$, its rest variables $\mathbf{r}_k$ and design variables $\mathbf{d}_k$ are a function of the part parameters $\mathbf{p}$ and the assignment $\mathbf{c}$. As a consequence, the equilibrium state $\mathbf{x}_k^*(\mathbf{d}_k)$ is a function of $(\mathbf{p}, \mathbf{c})$ as well. We therefore formulate the objective function for the kit of parts optimization as the sum of the design preservation energies across all designs:

$$\mathcal{J}(\mathbf{p}, \mathbf{c}) = \sum_k \mathcal{F}\left(\mathbf{x}_k^*(\mathbf{p}, \mathbf{c}), \mathbf{r}_k(\mathbf{p}, \mathbf{c})\right). \tag{5.3}$$

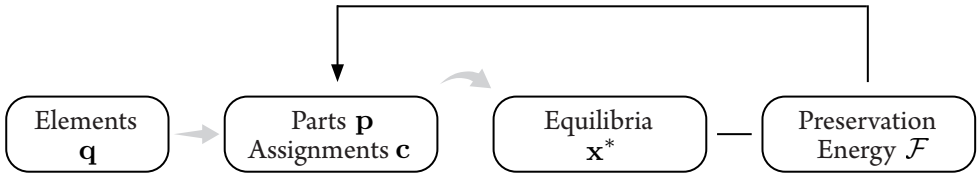


Figure 5.4: Optimization flow for the original kit of parts problem. The objective is to find the optimal part parameters $\mathbf{p}$ and assignments $\mathbf{c}$ that minimize the design preservation energy $\mathcal{F}$ across all designs.

Figure 5.4 illustrates the optimization problem. Optionally, weights can be assigned to indicate the relative importance of each design and re-formulate the objective as a weighted sum of design preservation energies.

5.5 Kit of Parts Optimization

The equilibrium state $\mathbf{x}^*$ in Equation (5.3) is sensitive to changes in the kit-of-parts variables $(\mathbf{p}, \mathbf{c})$. A change in the assignment function can lead to a large jump in the equilibrium state $\mathbf{x}^*$ and subsequently the design preservation energy $\mathcal{J}$. In addition, the space of assignments $\mathbf{c}$ grows exponentially with n, making an exhaustive search over the m^n possibilities intractable. The projection of elements $\mathbf{q}$ onto parts $\mathbf{p}$ in the context of bending-active structures is challenging and can result in buckled equilibrium states. We discuss more about buckling issues in Section 5.8 and Figure 5.11 illustrates how our approach mitigates them.

Projection-Relaxed Problem

We address the forementioned challenges by formulating a relaxation of the problem of minimizing Equation (5.3) to a tractable continuous optimization. This relaxation is achieved by tracking auxiliary continuous variables $\tilde{\mathbf{q}}$ of the elements in the simulation.

We define the kit-of-parts parameters $(\mathbf{p}, \mathbf{c})$ as dependent variables of $\tilde{\mathbf{q}}$ and introduce a projection energy $\mathcal{P}$ to bind the auxiliary variables $\tilde{\mathbf{q}}$ to the parts and assignment variables $(\mathbf{p}, \mathbf{c})$. The rest variables $\mathbf{r}_k$ and the equilibrium state $\mathbf{x}_k^*(\mathbf{r}_k)$ of design S_k are defined as functions of $\tilde{\mathbf{q}}$. The new objective is then written as a function of $\tilde{\mathbf{q}}$:

$$\mathcal{J}(\tilde{\mathbf{q}}) = \sum_k \mathcal{F}\left(\mathbf{x}_k^*(\tilde{\mathbf{q}}), \mathbf{r}_k(\tilde{\mathbf{q}})\right) + \mathcal{P}(\tilde{\mathbf{q}}). \tag{5.4}$$

As opposed to the former objective of Equation 5.3, part parameters $\mathbf{p}$ and part assignments $\mathbf{c}$ are dependent variables of $\tilde{\mathbf{q}}$ and are updated in the optimization loop. The relaxed formulation de-

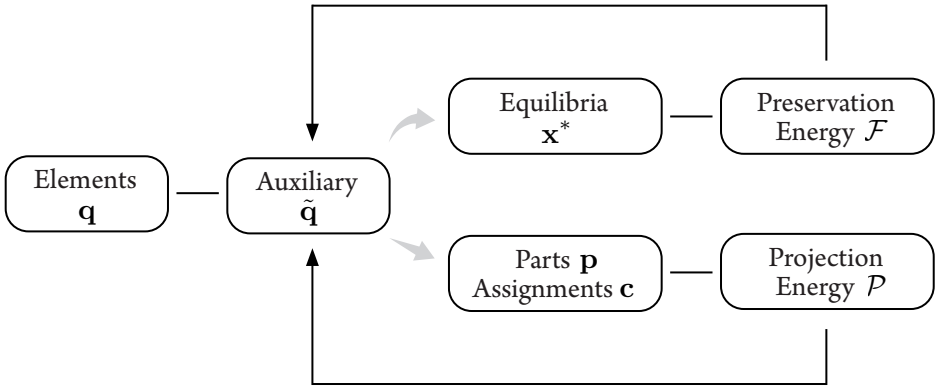


Figure 5.5: Optimization flow for the relaxed problem. The relaxation enables a fully differentiable optimization that can be seamlessly combined with a physics-based simulation. The elements $\mathbf{q}$ and parts $\mathbf{p}$ are illustrated in Figure 5.6. Figure 5.7 visualizes the part assignments $\mathbf{c}$ and equilibria $\mathbf{x}^*$ along with the energies $\mathcal{F}$ and $\mathcal{P}$.

fines the equilibrium simulation as a function of the continuous variables $\tilde{\mathbf{q}}$ and makes $\mathcal{J}$ robust to large jumps due to changes in the assignment function. The optimization flow is illustrated in Figure 5.5. The various terms in $\mathcal{J}(\tilde{\mathbf{q}})$ specific to our bending-active systems are defined in Section 5.6 and illustrated in Figure 5.7 on a single design instance for each system.

Projection Energy

To define $\mathcal{P}$, we first introduce a non-dimensionalized part-element projection energy $\rho : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$. A low value of $\rho(\mathbf{p}_i, \tilde{\mathbf{q}}_j)$ indicates higher similarity between an element j and a part i. For independent parts $\mathbf{p}$, assignment $\mathbf{c}$, and elements $\tilde{\mathbf{q}}$, the projection energy $\tilde{\mathcal{P}}(\mathbf{p}, \mathbf{c}, \tilde{\mathbf{q}})$ is aggregated over all elements as,

$$\tilde{\mathcal{P}}(\mathbf{p}, \mathbf{c}, \tilde{\mathbf{q}}) := \frac{w_\mathrm{c}}{n} \sum_{j=1}^{n} \rho(\mathbf{p}_{c(j)}, \tilde{\mathbf{q}}_j), \tag{5.5}$$

where the weight w_c controls the relative importance given to the projection energy term during optimization. Minimizing $\tilde{\mathcal{P}}$ ensures that the elements are well represented by the parts they have been assigned.

The projection energy $\mathcal{P}(\tilde{\mathbf{q}})$ from Equation (5.4) is then obtained as a minimum over all possible parts and assignments for a given set of elements $\tilde{\mathbf{q}}$,

$$\mathcal{P}(\tilde{\mathbf{q}}) := \min_{\mathbf{p},\mathbf{c}} \tilde{\mathcal{P}}(\mathbf{p}, \mathbf{c}, \tilde{\mathbf{q}}). \tag{5.6}$$

We minimize $\tilde{\mathcal{P}}$ in an alternating fashion over assignment and part updates.

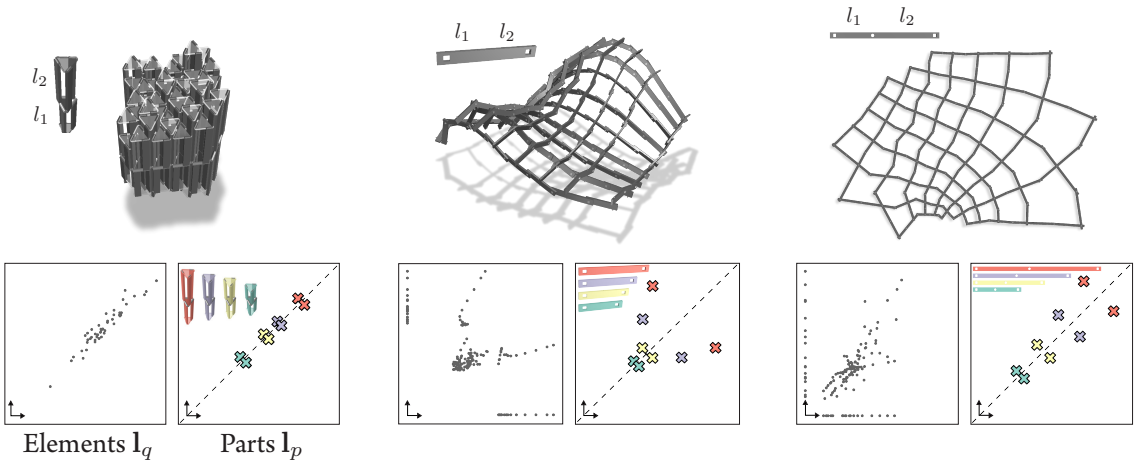


Figure 5.6: Elements, parts, and assembly states of the three classes of bending active structures we consider. From top to bottom: Umbrella Meshes, orthogonal grids, C-shells. The plots on the right illustrate the distribution of parameters in the input models compared to an optimized kit of parts, where element symmetries can be exploited to further reduce the number of parts.

Updates

The assignment $\mathbf{c}$, or the function c, is updated by keeping $\mathbf{p}$ fixed and solving for the optimal assignment,

$$c(j) := \operatorname*{argmin}_i \rho(\mathbf{p}_i, \tilde{\mathbf{q}}_j). \tag{5.7}$$

The parts $\mathbf{p}$ are then updated by keeping $\mathbf{c}$ fixed and solving for the optimal parts,

$$\mathbf{p}_i(\tilde{\mathbf{q}}) := \operatorname*{argmin}_{\mathbf{y}\in\mathbb{R}^d} \sum_{j\in c^{-1}(\{i\})} \rho(\mathbf{y}, \tilde{\mathbf{q}}_j), \tag{5.8}$$

where $c^{-1}(\{i\})$ is the set of elements currently assigned to part i. In all of our examples, the update step in Equation (5.8) can be solved analytically and efficiently differentiated. The alternating updates are repeated until convergence for every evaluation of $\mathcal{P}(\tilde{\mathbf{q}})$. Note that when ρ is the squared L^2 distance, the update rules are equivalent to the k-means clustering algorithm [MacQueen et al., 1967].

Initialization

The alternating update scheme to compute $\mathcal{P}(\tilde{\mathbf{q}})$ requires an initial guess for $(\mathbf{p}, \mathbf{c})$. Our initialization is inspired by the k-means++ algorithm [Arthur et al., 2007], so that parts are as spread out over the set of elements as possible. A first element is chosen uniformly at random and assigned to the first part $\mathbf{p}_1$. A new element $\tilde{\mathbf{q}}_j$ is chosen at random among the remaining elements

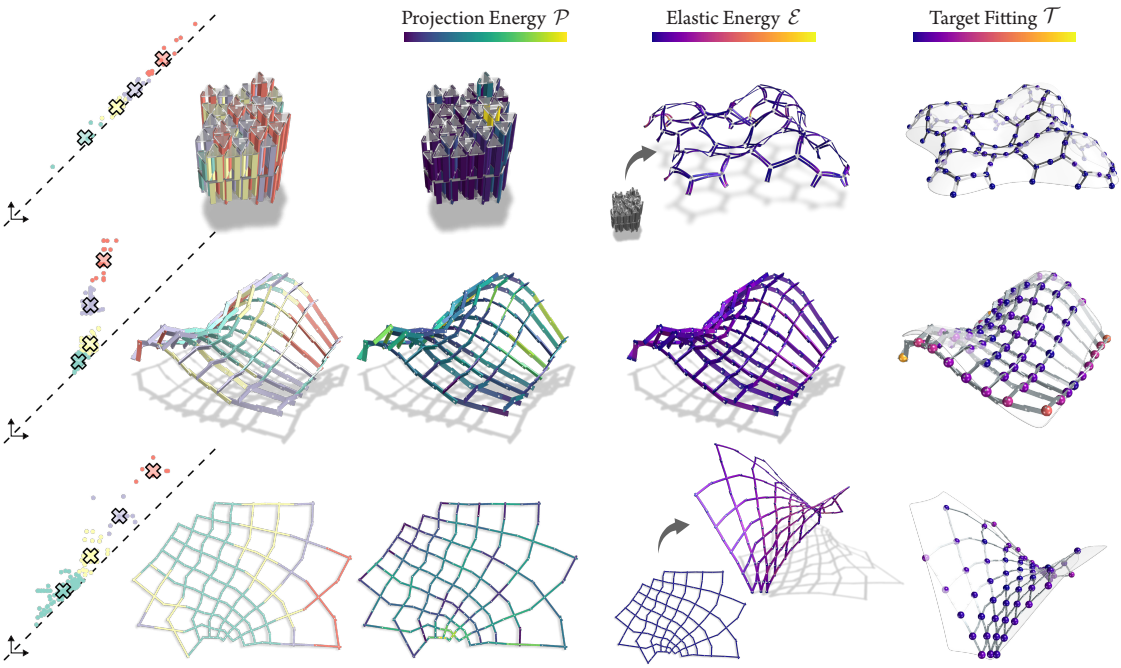


Figure 5.7: Optimization objectives in the relaxed problem: For a given input design, the objective function $\mathcal{J}$ is composed of the design preservation energy $\mathcal{F}$ $(= \mathcal{T} + \mathcal{E})$ and the projection energy $\mathcal{P}$. The plots on the left show distributions of element parameters $\tilde{\mathbf{q}}$ and the optimal part assignments $(\mathbf{p}, \mathbf{c})$ minimizing the projection energy $\mathcal{P}$. The dots and crosses represent $\tilde{\mathbf{q}}$ and $\mathbf{p}$ as defined in Figure 5.6. The subsequent columns illustrate each of the terms in $\mathcal{J}$ for the three classes of bending-active structures.

according to a probability proportional to the squared distance $\min_i \rho(\mathbf{p}_i, \tilde{\mathbf{q}}_j)^2$, where i spans the initialized parts and j indexes the remaining unassigned elements. The chosen element j is then assigned to part i and i is added to the set of initialized parts. This process is repeated until m parts have been constructed. We observe in Figure 5.7 (left) that such a process allows the parts to span the element space well.

5.6 Specialization to Bending-active Systems

We specialize our general computational pipeline to optimize a kit of parts for the chosen bending-active systems: Umbrella Meshes [Ren et al., 2022], Orthogonal Grids, and C-shells [Becker et al., 2023]. Physics-based simulations of the involved bending-active structures are based on the methods presented in the respective papers which have been validated by fabricating prototypes.

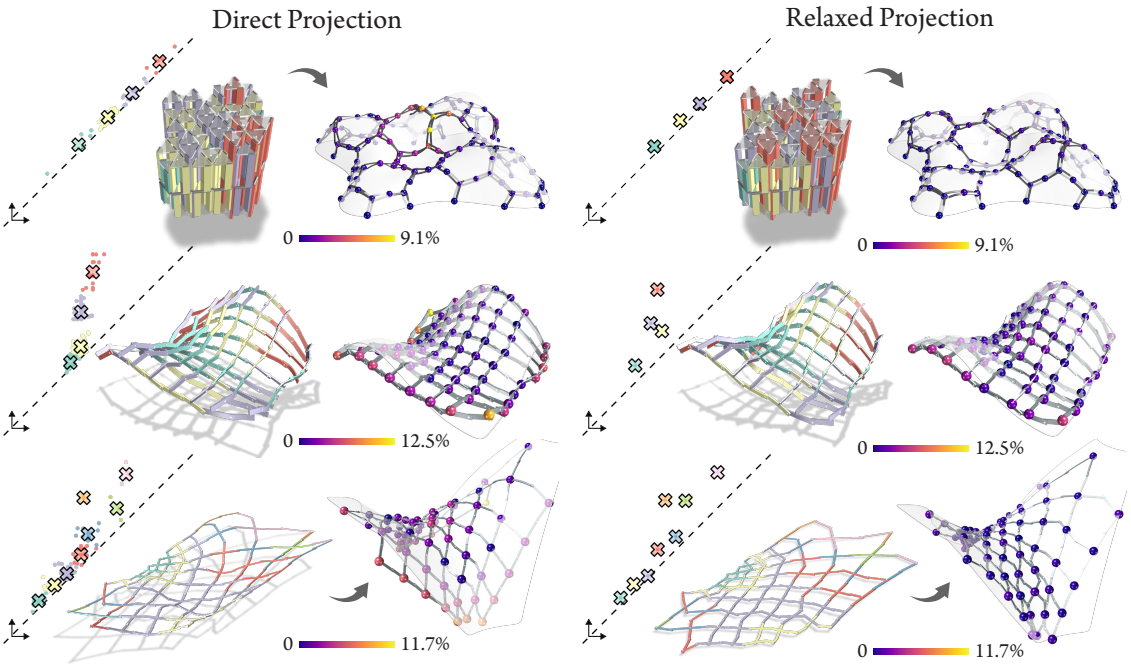


Figure 5.8: The parts $\mathbf{p}(\mathbf{q})$ and assignments $\mathbf{c}(\mathbf{q})$ directly extracted from the original design yields poor preservation of the target shape (left). By re-arranging and collapsing elements around their assigned parts, our relaxed optimization process allows for a better preservation of the initial design (right).

We formulate the objective from Equation (5.3) specific to these three systems. In particular, we define the elements and parts parameterizations, the deployment process, and the element-to-part projection energy. The datasets of shapes for all systems are shown in the supplemental material.

Umbrella Meshes

Umbrella Meshes are composed of modular volumetric scissor linkages, coined *umbrella cells*. Each umbrella cell deploys from a compact vertical configuration to a flat expanded state whose footprint depends on the height of the cell. When umbrella cells of different heights are assembled together and deployed, metric frustration due to expansion incompatibilities causes the structure to deform into a doubly-curved bending-active surface structure. The top corner of Figure 5.6 illustrates the elements that make up an umbrella cell, which can be rationalized into a kit of parts for subsequent assembly and deployment. The plates, X-joints and T-joints are identical across all the designs.

Preservation Energy

Each umbrella cell is defined in its rest state by the lengths of the arms connected to the top plate (top heights) and the lengths of the arms connected to the bottom plate (bottom heights). [Ren et al., 2022] explain how different top and bottom heights lead to programming mean curvature in the deployed state. The rest variables $\mathbf{r}$ are defined as the aggregation of these lengths. The design variables $\mathbf{d}$ may also include pinned vertices at the boundary as for model 2 in Figure 5.11. The structure is deployed in a similar strategy to the one described in [Ren et al., 2022]: Top and bottom plates are brought together through a linear actuator and rigid motions are pinned using small surface attraction forces when there are no boundary constraints. These additional conservative forces responsible for deployment are modeled by the energy term $\mathcal{D}$ in Equation (5.1). The resulting deployed state $\mathbf{x}^*$ is then used to evaluate the objective $\mathcal{J}$ in Equation (5.4).

Projection Energy

Each umbrella unit/element, is parameterized by the length of the arms of the top plate and the bottom plate, $\mathbf{l}_q \in \mathbb{R}^2_+$. Naturally, we define parts as two lengths $\mathbf{l}_p \in \mathbb{R}^2_+$. Parts can be used as is or in a mirrored configuration, which is illustrated by the mirror symmetry about the bisector of the first quadrant in Figure 5.6 (top right). The projection energy between elements and parts is defined as the squared L^2 norm of the difference between the two lengths in both part configurations,

$$\rho(\mathbf{l}_p, \mathbf{l}_q) = \frac{1}{2l_\rho} \min\left(\|\mathbf{l}_p - \mathbf{l}_q\|_2^2, \|\text{flip}(\mathbf{l}_p) - \mathbf{l}_q\|_2^2\right), \tag{5.9}$$

where $\text{flip}(\mathbf{p}_i)$ flips the lengths of the part $\mathbf{p}_i$, and l_ρ is chosen as the median of the arm lengths across the input designs. In this case, the update rule in Equation (5.8) can be solved analytically by sorting the assigned elements' lengths and taking the mean. Note that the sorting indices need to be stored to ensure the correct assignment of the parts to the elements.

Orthogonal Grids

Orthogonal grids are bending-active structures composed of elastic lamellas attached at the crossings. The lamellas are oriented such that their strong axis is orthogonal to the input design surface. C-meshes [Liu et al., 2023a] shown in Figure 5.2 are a special case of orthogonal grids that further imposes the structure to be collapsible into a flat state. In general, orthogonal grids require curved elements to best approximate the underlying target surface. We rationalize the curved rods as piecewise straight beams rigidly connected at their *corners*. Exactly one corner is allowed

between two neighboring joints along each rod, to spatially separate inner- and inter-rod connections.

Preservation Energy

The design state $\mathbf{d}$ of an orthogonal grid comprises rod segment rest lengths together forming the rest variables vector $\mathbf{r}$, and corner angles. Inter-rod connections are simulated using rotational joints, effectively constraining the intersection point of two rods while allowing rotation. Corners, or inner-rod connections, are simulated as rigid joints with the opening angle between two rod segments treated as a fixed variable. The corner angles form the set of fixed simulation variables $\mathbf{x}_f$ (Dirichlet constraints) of the equilibrium problem in Equation (5.1). We rule out rigid motions of the structures during simulation using the same strategy as in Umbrella Meshes.

Projection Energy

Each straight element is parameterized by the distance from the first corner joint to the rotational joint and the distance from the rotational joint to the second corner as shown in Figure 5.6 (left). The order of the corners is given by following the curves in the original design. Boundary elements are distinguished from inner elements as they have only one corner joint, and are defined by a single length.

We consequently define each part as two lengths $\mathbf{l}_p \in \mathbb{R}^2_+$. Similar to parts in Umbrella Meshes, parts can be used as is or mirrored. The projection energy is defined equivalently to the Umbrella Meshes case, and the parts update rule is solved similarly using the sort strategy. We use the median of the rod lengths across the input designs as the reference length l_ρ in Equation (5.9).

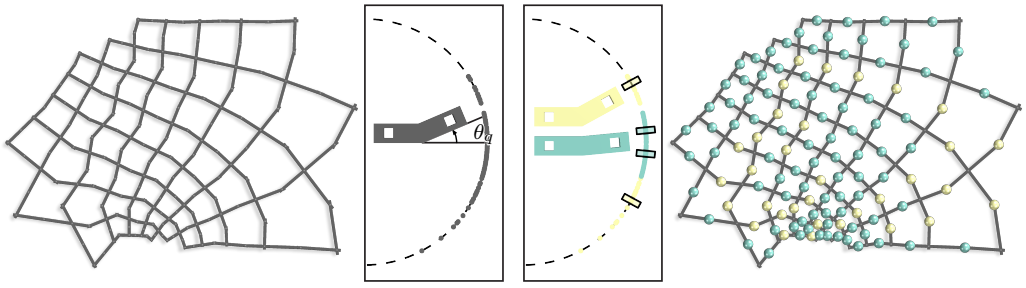


Figure 5.9: Angular elements $\theta_q \in [-\pi, \pi]$ are computed at the corners of rationalized C-shells. The angular part is defined by a single angle $\theta_p \in [0, \pi]$ and can be flipped. We represent the angular elements, the 2 parts and their flipped configurations on the unit circle.

To ease the assembly process, corner angles are also grouped into their own discrete set of parts. Our kit of parts is then composed of two types of parts: Straight beams (linear parts) and corner angles (angular parts). The corner angles are parameterized by the angle between the two straight beams they connect, as shown in Figure 5.9. Parts are defined subsequently by a single angle $\theta_p \in [0, \pi]$, and can be mirrored during assembly. Our projection energy is then defined as the squared difference between the two unsigned angles $\rho(\theta_p, \theta_q) = \frac{1}{2}\left(\theta_p - |\theta_q|\right)^2$. Based on that definition, the update rule in Equation (5.8) is the average of the assigned elements' unsigned angles.

C-shells

C-shells are deployable gridshells composed of curved elastic beams connected through rotational joints [Becker et al., 2023]. The assembly state is stress-free by definition and the structure is deployed via torque actuation, by constraining the average opening angle at the rotational joints. For a completely custom fabricated C-shell tailored to deploy to a specific surface, the curved beams are laser cut precisely following some optimized splines. Similar to Orthogonal Grids, we rationalize each of the curved beams using piecewise straight beams joined at rigid corners. Rotational and corner joints now share the same axis in the rest state.

Preservation Energy

Each rationalized C-shell is defined by the lengths of the straight beams, the corner angles, and the average opening angle of the single-axis rotational joints in the deployed state. These design variables are aggregated in $\mathbf{d}$. Rigid connections at the corners are simulated using Dirichlet constraints in Equation (5.1). Depending on the values of beam lengths and corner angles, a rationalized C-shell no longer has a guaranteed zero-energy *rest equilibrium state* when assembled. We therefore compute this state $\mathbf{x}_\mathrm{r}^*$ by solving an equilibrium problem for the assembly configuration. The deployed equilibrium state $\mathbf{x}_\mathrm{d}^*$ is obtained by further constraining the average opening angle at the rotational joints as described in Becker et al. [2023]. In order to mitigate incompatibilities in the rest state $\mathbf{x}_\mathrm{r}^*$ and ease assembly, we incorporate the energy of the rest equilibrium state in the preservation energy as

$$\mathcal{F}\left(\mathbf{x}_\mathrm{r}, \mathbf{x}_\mathrm{d}, \mathbf{r}\right) = \mathcal{T}\left(\mathbf{x}_\mathrm{d}\right) + \mathcal{E}\left(\mathbf{x}_\mathrm{d}, \mathbf{r}\right) + \mathcal{E}\left(\mathbf{x}_\mathrm{r}, \mathbf{r}\right), \tag{5.10}$$

and we update $\mathcal{J}$ to track both equilibria as

$$\mathcal{J}(\tilde{\mathbf{q}}) = \sum_k \mathcal{F}\left(\mathbf{x}_{\mathrm{r},k}^*(\tilde{\mathbf{q}}), \mathbf{x}_{\mathrm{d},k}^*(\tilde{\mathbf{q}}), \mathbf{r}_k(\tilde{\mathbf{q}})\right) + \mathcal{P}(\tilde{\mathbf{q}}), \tag{5.11}$$

where k indexes the different designs in the input set.

Projection Energy

We define each linear part as two lengths $\mathbf{l}_p \in \mathbb{R}^2_+$, and each corner angle as a single angle $\theta_p \in [0, \pi]$. The projection energies between elements and parts of the same type, and the parts' update rules are defined identically to the Orthogonal Grids case.

5.7 Two-Stage Optimization

We first show a direct rationalization approach in Figure 5.8 that computes parts $\mathbf{p}$ and assignments $\mathbf{c}$ directly from the input elements $\mathbf{q}$ by minimizing $\tilde{\mathcal{P}}(\mathbf{p}, \mathbf{c}, \mathbf{q})$. This approach does not take the design preservation energy $\mathcal{F}$ into account and can lead to undesired buckling in the output designs. Then, we illustrate the performance of the unrelaxed approach from Section 5.4 which optimizes the part parameters $\mathbf{p}$ using the assignments from minimizing $\tilde{\mathcal{P}}(\mathbf{p}, \mathbf{c}, \mathbf{q})$ directly. We then compare our relaxed approach in Figure 5.11. We propose a two-stage optimization approach that first optimizes the relaxed problem and then fine-tunes the parts $\mathbf{p}$ while keeping the assignments $\mathbf{c}$ fixed to minimize the total preservation energy $\mathcal{F}$. We show the results of our approach in Figure 5.12.

First Stage: Relaxation

The original design rationalization problem described in Section 5.4 is a combinatorial optimization problem, where the number of possible part-to-element assignments grows as m^n. This makes the problem intractable for large elements count n and a non-trivial kit of parts.

Instead, our relaxation allows leveraging tools from the continuous optimization literature to efficiently solve

$$\tilde{\mathbf{q}}^* := \underset{\tilde{\mathbf{q}}}{\operatorname{argmin}}\, \mathcal{J}(\tilde{\mathbf{q}}), \text{ s.t. } \mathbf{p}_{\min} \leq \tilde{\mathbf{q}} \leq \mathbf{p}_{\max}, \tag{5.12}$$

where $\tilde{\mathbf{q}}$ now contains all elements of different kinds (linear and angular if applicable), and $\mathbf{p}_{\min}$ and $\mathbf{p}_{\max}$ are lower and upper bounds. These are derived from the parts feasibility and fabricability constraints e.g., for the minimum distance between corners and joints for linear parts. We assume that elements of the same kind share the same constraints. In the supplemental material, we show that for all the part-element projection energies ρ we use in our experiments, the resulting parts obtained from the optimal elements $\mathbf{p}(\tilde{\mathbf{q}}^*)$ using the update rule in Equation (5.8) are *guaranteed* to satisfy the original feasibility constraints. This fact effectively positions our method

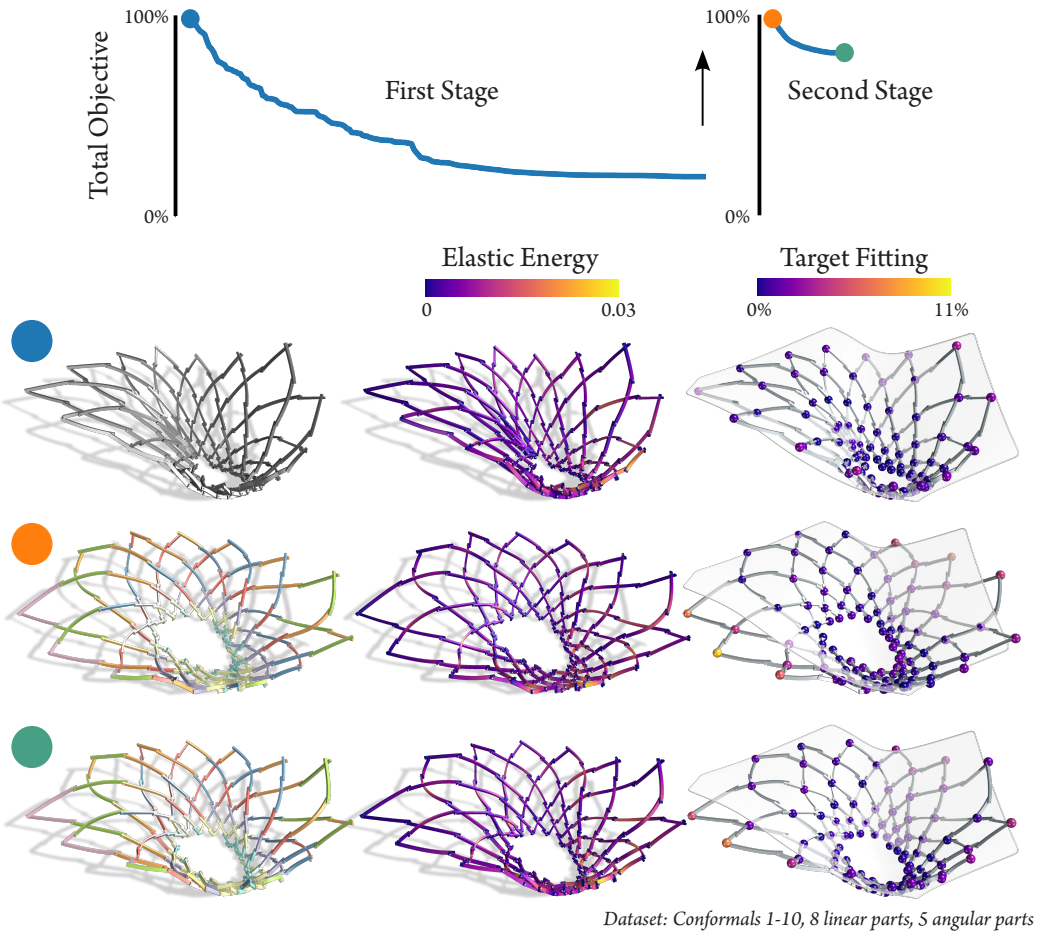


Figure 5.10: Our two stage optimization fine-tunes designs after first optimizing the relaxed problem. The arrow indicates the projection of the solution of the first stage $\tilde{\mathbf{q}}^*$ onto the part set to obtain $\mathbf{p}(\tilde{\mathbf{q}}^*)$, the initial guess for the second stage. We plot the objective of each stage normalized with respect to the initial value for that stage. The rows show the designs at the start, between the two stages, and at the end of the full optimization process.

as a co-rationalization approach providing an end-to-end parametric control over the output designs with respect to the part feasibility constraints.

Figure 5.11 shows how our projection onto a kit of parts after our first-stage relaxation preserves designs better compared to directly solving the original unrelaxed problem. This forms the base for the second stage of our optimization.

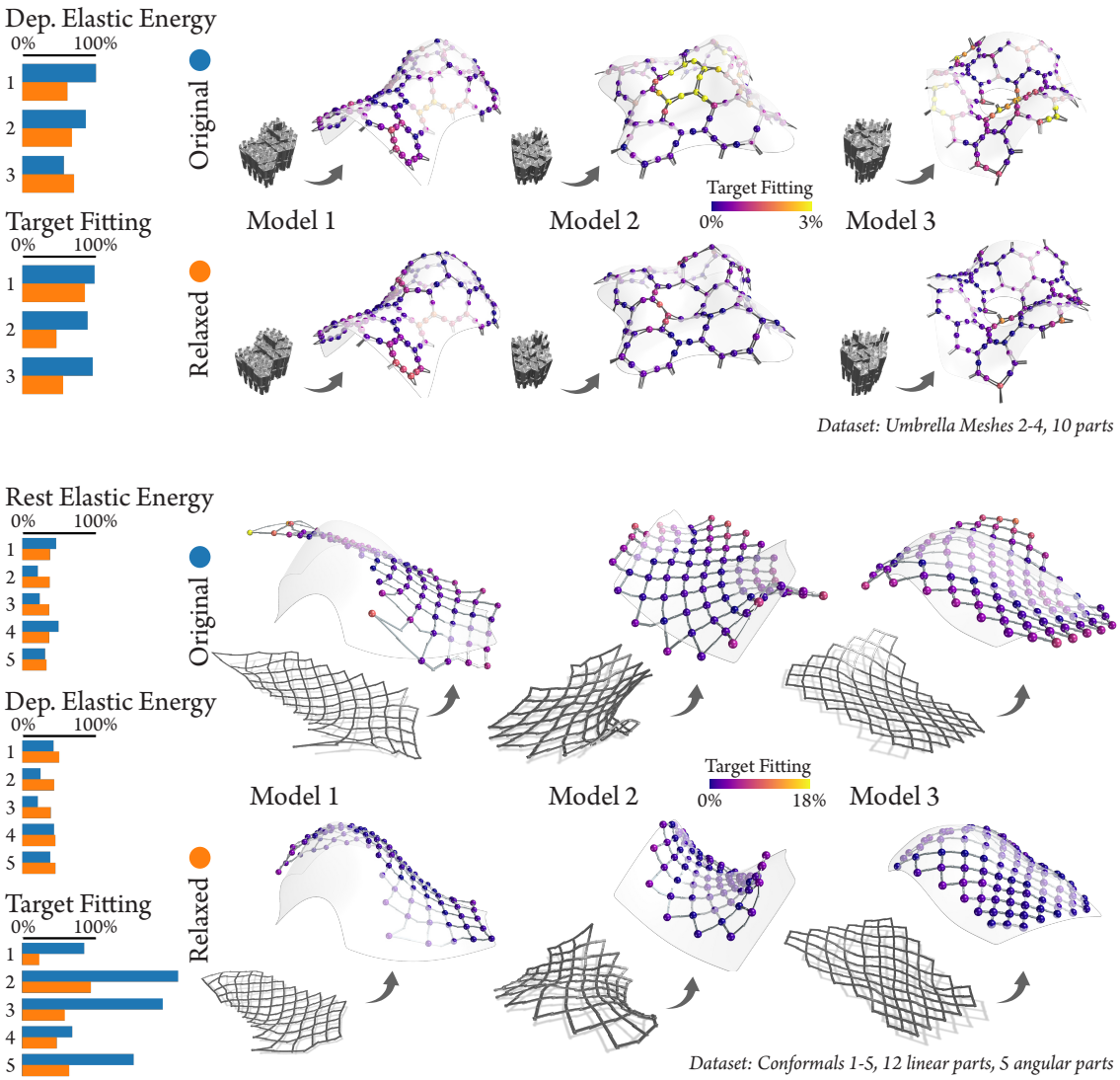


Figure 5.11: Solving for the original minimization problem over the parts $\mathbf{p}$ in Equation (5.3) using the original assignments $\mathbf{c}(\mathbf{q})$ produces undesired buckling. Our relaxation allows smoothly bringing designs together towards a shared kit of parts. We show designs after the first stage optimization involving the relaxation and project the elements on to the parts at the end of it. We report target fitting as a percentage of each model's bounding box diagonal. The optimization quantities are normalized using the respective initial values for each "projected" design.

Second Stage: Fixed Assignment Fine-Tuning

The first stage optimization strives to concentrate all the *auxiliary elements* tightly around the *parts* in order to mitigate the discontinuous elements-to-parts conversion. However, the output goal is still a part set, and thus the auxiliary variables need to be converted to the assigned parts. To this

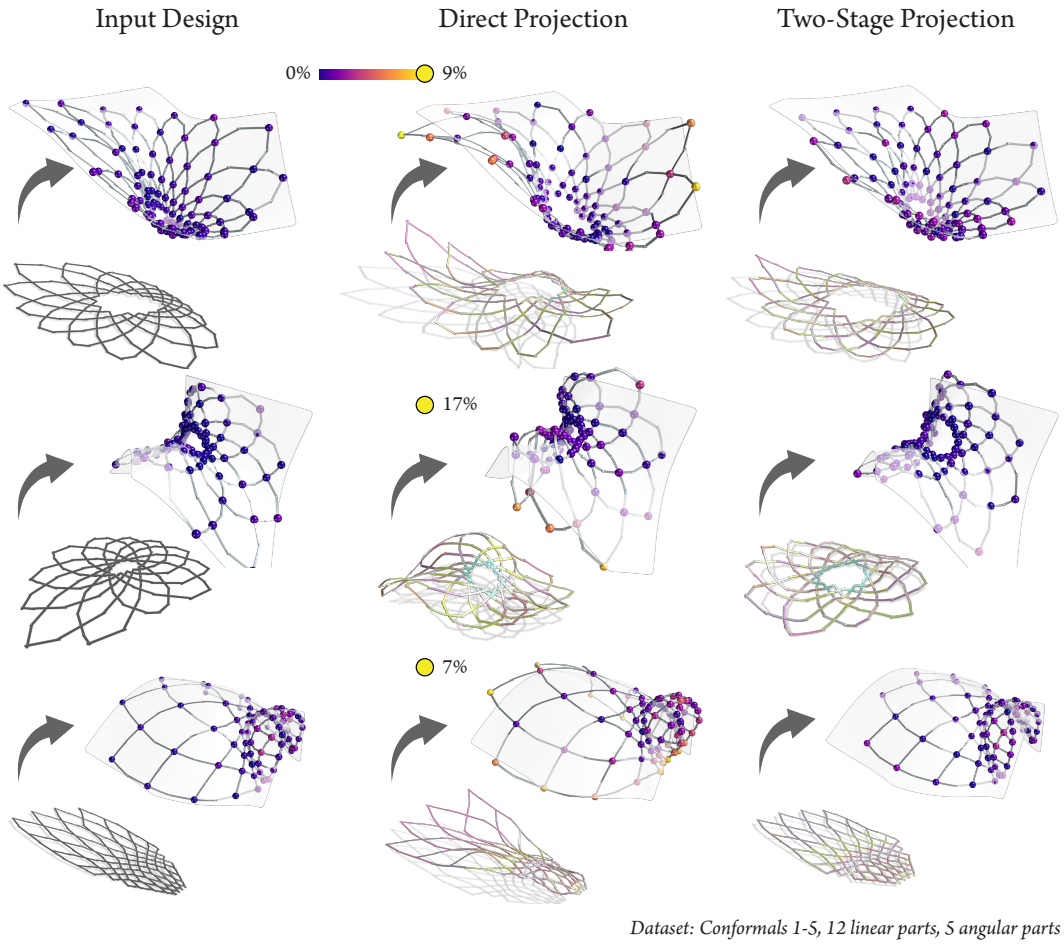


Figure 5.12: We show subsets of optimized designs of discretized C-shells after the two-stage optimization process. The left column shows the input designs, the middle column shows the designs directly projected to the part set. The right column shows the designs after the two-stage optimization process. We report the target fitting as a percentage of each model's bounding box diagonal.

end, we perform a second stage of fine-tuning, where we fix the assignment $\mathbf{c}(\tilde{\mathbf{q}}^*)$ and optimize the part parameters starting from $\mathbf{p}(\tilde{\mathbf{q}}^*)$, as

$$\mathbf{p}^* := \underset{\tilde{\mathbf{q}}}{\operatorname{argmin}}\, \mathcal{J}\left(\mathbf{p}, \mathbf{c}(\tilde{\mathbf{q}}^*)\right), \text{ s.t. } \mathbf{p}_{\min} \leq \mathbf{p} \leq \mathbf{p}_{\max}. \tag{5.13}$$

Here $\mathbf{p}_{\min}$ and $\mathbf{p}_{\max}$ are the lower and upper bounds of the parts parameters, and the objective function $\mathcal{J}$ is defined as in Equation (5.3). The second stage optimization has a very low number of variables compared to the first stage since the assignments are fixed and the part set is sparse. The fine-tuning explores the part space locally within a specific assignment for a local minimizer

of the objective function $\mathcal{J}$. Figure 5.10 shows the two stage optimization process and visualizes objectives on an associated design instance. The transition between the two stages often leads to a jump in the elastic energy which is mitigated during the second stage. Figure 5.12 shows the final results of the two stage optimization process for C-shells and Umbrella Meshes and compares them to the input designs as well as a direct rationalization approach.

Optimization Algorithm

We solve the optimization problems in Equation (5.12) and Equation (5.13) using a trust-region active-set method (Sequential Linear-Quadratic Programming) with a BFGS Hessian approximation [Nocedal and Wright, 2006] provided by Knitro [Waltz and Nocedal, 2004]. We compute the gradient of the preservation energy $\mathcal{F}$ using first order adjoint sensitivity analysis. More details on how differentiation with respect to the constrained simulation variables $\mathbf{x}_f^{\mathrm{tgt}}(\mathbf{d})$ is performed can be found in Appendix B.

We use uniform material properties for all the structures in our experiments with Young's modulus E and Poisson's ratio ν set to $(1400\mathrm{MPa}, 0.35)$ for Umbrella Meshes, and $(2100\mathrm{MPa}, 0.35)$ for Orthogonal Grids and C-shells. Our first stage optimization typically converges in 10 minutes to an hour on the examples we show. The second stage optimization runs faster, and typically takes no more than 15 minutes to converge in our experiments.

5.8 Discussion

Projection onto an Existing Kit of Parts

We demonstrate in Figure 5.13 how our relaxation approach can be adapted to find the best assembly that approximates an input design using only an existing kit of parts $\bar{\mathbf{p}}$. We jointly optimize the assignment $\mathbf{c}$ and the element parameters $\tilde{\mathbf{q}}$ for the objective of the relaxed optimization problem shown in Equation (5.4). Since parts are fixed initially, the projection energy $\mathcal{P}$ in Equation (5.6) becomes $\min_{\mathbf{c}} \tilde{\mathcal{P}}(\bar{\mathbf{p}}, \mathbf{c}, \mathbf{q})$, where $\tilde{\mathcal{P}}$ is defined in Equation (5.5). As shown in Figure 5.13, a higher projection energy weight w_{c} enforces a tighter and faster fit of each element to their initial assignment, preventing them from evolving during the projection operation. Lowering projection energy weight w_{c} allows smoothly transitioning from the input design to the kit of parts, leading in practice to better assignments. However, if the weight is set too low, the parts are only loosely fit by the elements. The part assignments step may then lead to undesired buckling in the output design, as shown in Figure 5.13 for $w_{\mathrm{c}} = 10^{-5}$.

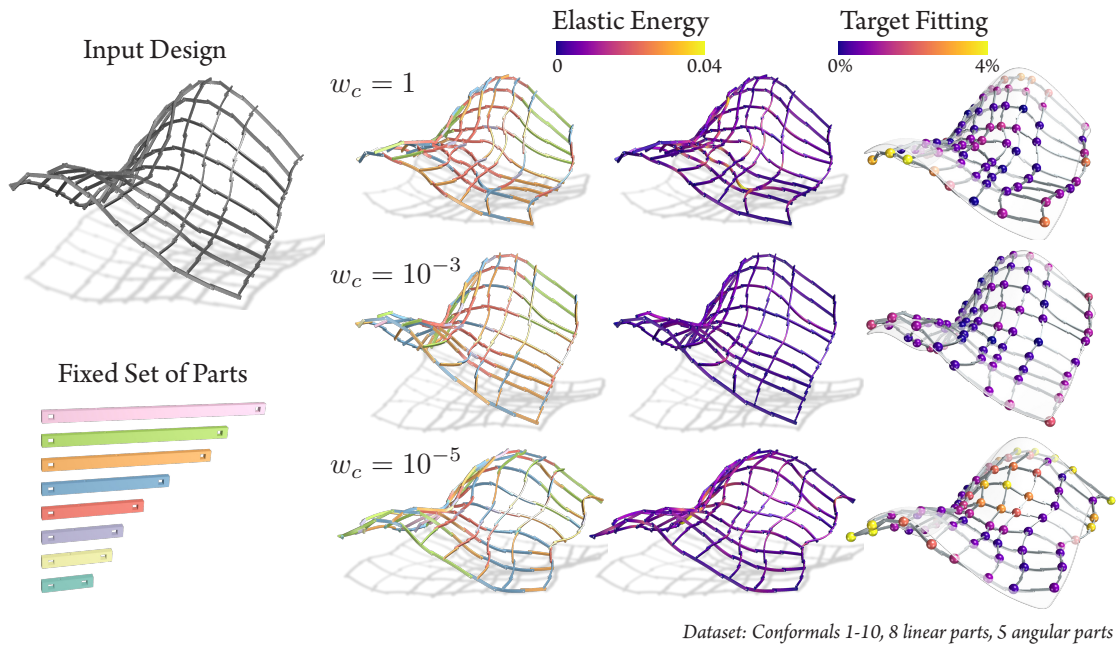


Figure 5.13: A new design is projected onto an existing kit of parts. The input design has not been used during the optimization of the kit of parts. Tuning the clustering weight allows recovering a better design.

Part Reuse

Consider the scenario where we fabricate only as many pieces (instances of parts) as needed such that all the designs can be realized individually using this kit of parts. When we use a small part set, the number of fabricated pieces is low, but the design fidelity can suffer due to the limited expressiveness of the parts. On the other hand, a large part set requires more pieces to be fabricated. Our framework can be used to investigate the trade-off between design fidelity and the size of the kit of parts. Figure 5.14 illustrates the interplay between the kit-of-parts size and the maximum deviation of the design instances from their target surfaces. In the visualized examples of rationalized designs, we observe clear improvement in the target fitting as the number of parts increase, however leading to an increase in the total number of fabricated pieces. Notice that the gain in target fitting diminishes after a certain number of parts (12 in Figure 5.14), suggesting that the additional parts do not contribute significantly to the design fidelity. Thus, our framework enables discovery of the most suitable trade-off between design fidelity and fabrication complexity.

With more parts, the likelihood of reusing pieces across different designs decreases. Ideally, we would like to reuse the same pieces across different designs to maximize part reuse. For the five

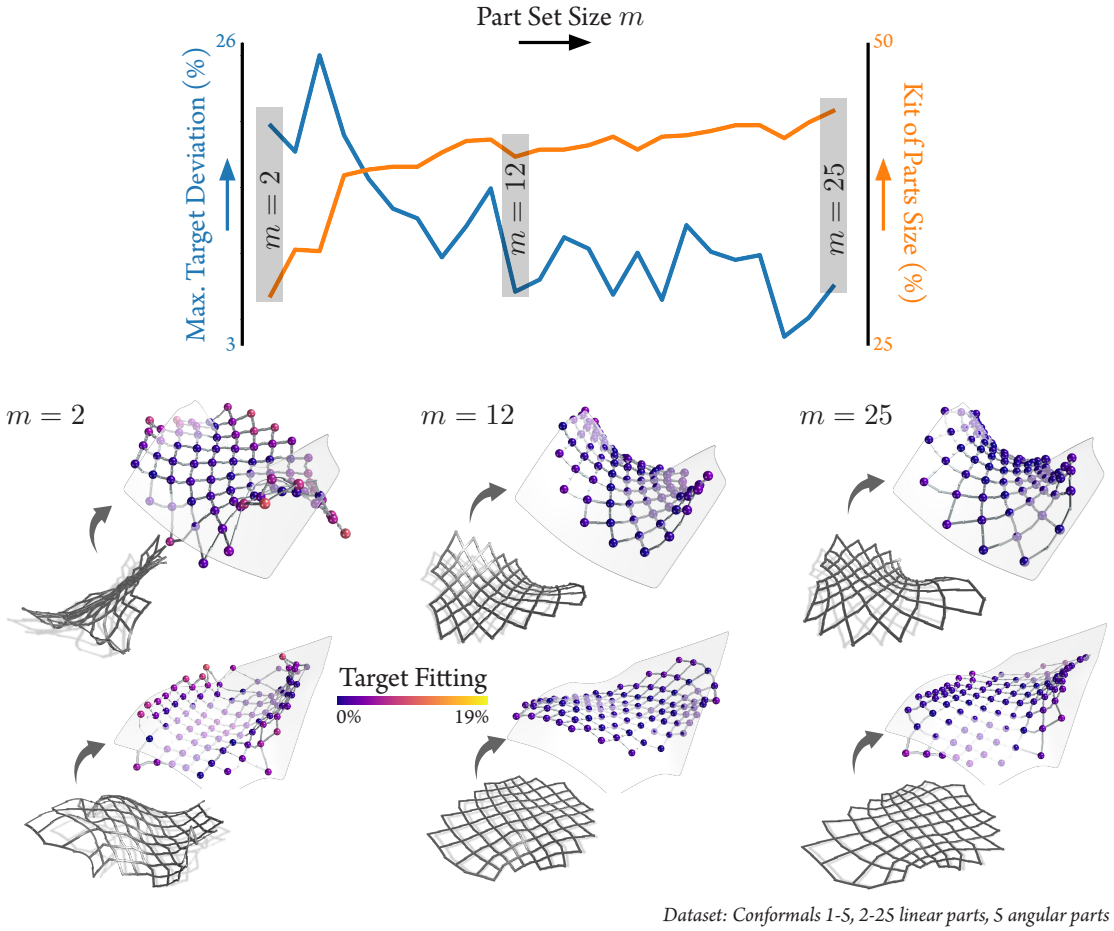


Figure 5.14: Increasing the number of distinct parts m trades design fidelity for parts reuse. We jointly optimize 5 designs using the relaxation and show the resulting equilibrium states (of 2 designs) after assigning parts to elements. The kit of parts size is shown as a percentage of the total number of fabricated pieces over the total number of elements in the input designs.

design instances considered in our experiments, we analyze part reuse among different subsets of designs for each part set size. For example, consider the sets of pieces $\mathcal{S}_1, \mathcal{S}_2$ used to assemble two designs respectively. Then the Jaccard similarity $|\mathcal{S}_1 \cap \mathcal{S}_2|/|\mathcal{S}_1 \cup \mathcal{S}_2|$ measures the part reuse between the two designs. Figure 5.15a shows the pairwise Jaccard similarities for the five design instances for two different part set sizes (2, 25). This notion can be extended to more than two designs by considering $|\bigcap_k \mathcal{S}_k|/|\bigcup_k \mathcal{S}_k|$. Visualizing part reuse among different subsets (Figure 5.15b) can provide insights into disconnected design subsets. A similar analysis can be done over the part subsets to discover disconnected part subsets in the kit of parts. As the number of distinct parts increases, the average part reuse diminishes, as shown in Figure 5.15c.

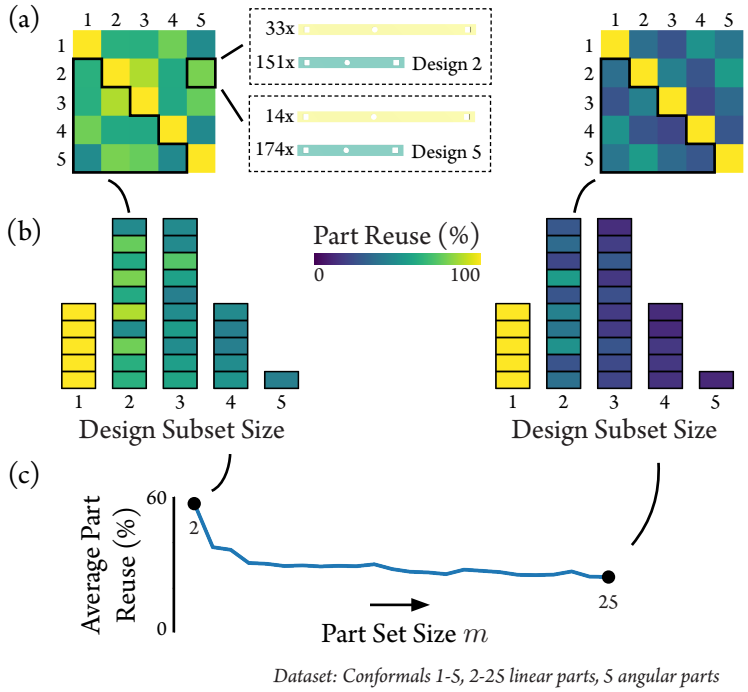


Figure 5.15: (a) Pairwise Jaccard similarities between two design instances for two different part set sizes (2, 25). The inset shows the pieces used in two designs (Design 2 and Design 5, with a part set size of 2) explaining the part reuse % in the incidence matrix measuring pairwise part reuse. (b) Generalized subset-wise part reuse. Note that the pairwise similarities correspond to similarities on subsets of cardinality 2. (c) Evolution of average part reuse across all subsets with the part set size.

Buckling Issues During Projection

Figure 5.8 shows how a direct projection of elements to parts can lead to undesired buckling even for a single design. Once such a bad assignment is made, optimizing the part parameters does not recover the design fidelity as seen in Figure 5.11. This is because the design objective is a highly non-linear, non-convex function of the rest variables. If elements are projected to parts when the clusters are not *tight* enough, the objective can incur a large jump sometimes manifested as the observed buckling. This problem is pronounced in the case of Umbrella Meshes which have multiple stable configurations owing to a larger design space, making recovery from buckled states hard.

Our relaxation provides a continuous balance between the objectives and brings the elements $\tilde{\mathbf{q}}$ closer to the parts $\mathbf{p}$ first while maintaining the design fidelity. The auxiliary variables $\tilde{\mathbf{q}}$ are then projected to the parts $\mathbf{p}$ at the start of the second stage optimization as described in Section 5.7.

Minimizing the relaxation objective $\mathcal{J}(\tilde{\mathbf{q}})$ in the first stage brings the elements $\tilde{\mathbf{q}}$ *tightly* close to the cluster center parts $\mathbf{p}$ controlled by w_{c}. While this is a significantly better state to project the elements to parts compared to the direct approach, it is unclear how *tight* the clustering should be. Owing to the highly nonlinear design objective, the equilibrium state can show large deviations for small changes in the element size resulting in failure cases like shown in Figure 5.18. In these scenarios, we increase the projection energy weight w_{c} before projection to explore better solutions without losing too much design fidelity. When that fails, we understand that the tight cluster of elements cannot be replaced by the cluster center part for this design, and re-initialize the clusters with a larger kit of parts.

Architectural Applications

As discussed in the introduction, bending-active structures are of particular interest in architecture. However, few designs have been realized, partly because of high cost and complexity of custom fabrication. The kit-of-parts approach that we propose can potentially alleviate these issues. In Figure 5.19 we show some speculative designs realized with our optimized kit of parts to illustrate the potential for applications in architecture.

Part Complexity

In our work, we show a kit-of-parts approach that first discretizes the input bending-active designs into atomic elements and then maps these elements to a part set that is parametrized the same way as each of the elements. We demonstrate results for three bending-active systems using this approach. However, our framework is general and can be extended to include more complex parts that are parametrized differently from the elements. For example, we can include parts that allow continuous adjustments.

Figure 5.16 shows an example of such more complex parts. Each straight beam has a slit window in which it can be connected to its neighboring elements. With the slits on both ends, the same part could then be used to replace multiple different elements. However, a compromise on the slit length must be found to preserve the structural integrity of the designs. Similarly, a reconfigurable umbrella part [Kusupati et al., 2023] can be used to replace multiple umbrella cells of different arm lengths that fall in the reconfigurable range. Reconfigurable elements provide additional degrees of freedom to find better solutions for the constrained problem of finding an optimal kit of parts. While complex reconfigurable parts have a greater degree of expressiveness, they also increase the fabrication complexity and can make assembly more difficult. For example,

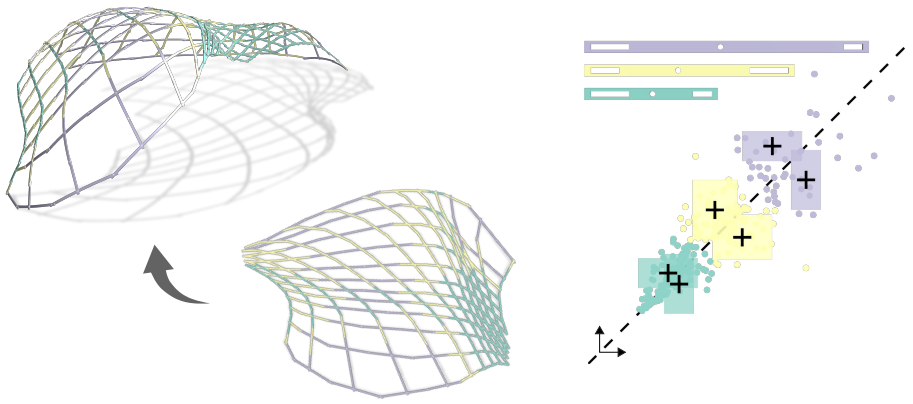


Figure 5.16: Our framework can accommodate optimizing more complex part typologies. Here we extend Orthogonal Grids' and C-shells' linear parts to include slits at the ends and allow realizing several element lengths by placing the *corner* at any point along the slit.

for the slitted parts shown here, parts need to be connected carefully to ensure the slit windows align correctly. In addition, the reconfigurable parts have additional components that can lead to collisions during deployment, or undesired aesthetics. We leave the thorough exploration of these reconfigurable part types for future work.

Computation Time

Figure 5.17 shows that most of the computation time is allocated to finding equilibria, which happens during the linesearch phase of the optimization. The gradient computation is done once per step and is relatively fast.

The optimization steps taken in the unrelaxed problem are on average more costly mainly due to a higher number of linesearch steps. The reduction from 2217 element variables to 29 part variables can explain that discrepancy. A perturbation in one of the part variables may impact more than 75 elements on average at once, in different ways depending upon their locations in their respective structures. The objective in the second stage appears more sensitive to each variable, making the optimization problem harder to solve.

The relaxed problem, on the other hand, requires fewer linesearch steps. The relaxation allows the elements to update independently and tracks a better-behaved equilibrium state.

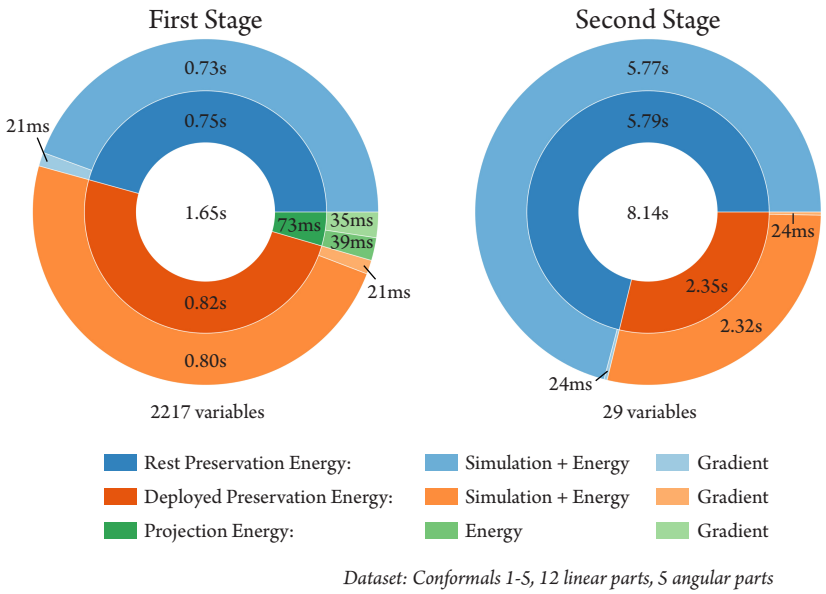


Figure 5.17: Timings of our two-stage optimization, reported per each successful line search step. The simulation time consists of solving the equilibrium problem possibly multiple times during the linesearch phase of the optimization. Gradient computation relies on the simulation output and is done once per step. For this experiment, the first stage took 1019 steps, and the second stage took 42 steps. Timings are measured on a Linux workstation with a 64-Core AMD Ryzen Threadripper 3990X Processor and 128GB of RAM.

5.9 Limitations and Future Work

Buckling Mitigation

While the relaxation approach performs well in our experiments, we currently cannot quantify how close the solution of the relaxed version is to the optimal part set and assignment. Depending on the energy landscape of the equilibrium that is being tracked, a small perturbation in the element parameters can lead to a significant change in the objective $\mathcal{J}$. When the solution from the relaxation problem is such an equilibrium state, we observe a jump (buckling) between the two stages of the optimization, which leads to potentially irrecoverable deployed state. Figure 5.18 exemplifies this. Since our optimization relies on local sensitivity information we cannot easily predict when such a jump will occur.

Focused Objectives

When optimizing the part geometries, we currently do not directly control re-use efficiency i.e., the number of parts shared among different designs. If structures are to be assembled in sequence,

it would be desirable if the next design could re-use as many parts from the previous design as possible to limit the total number of elements that need to be fabricated. A corresponding objective can be integrated into our optimization in future work.

While we currently use an elastic energy term in the design preservation to favor low-energy designs, we can also include focused objectives like the maximum stress in the structure. Similarly, precise load-bearing behavior can be enforced by including external loads as part of the equilibrium simulation. Incorporating these specific scenarios into our framework remains a future work.

Input Design Harmonization

Within each dataset, our input structures are designed such that initial elements belong to the same region of the element parameters space. This manual harmonization pre-processing step is necessary to ensure that the downstream optimization converges to parts shared across most designs. Automatically diagnosing the quality of the input designs and providing feedback e.g., in the form of rescaling or topology changes, on how to update the input designs best to make them more compatible with each other can be a valuable addition. This can keep users informed of the re-use efficiency during early design stages.

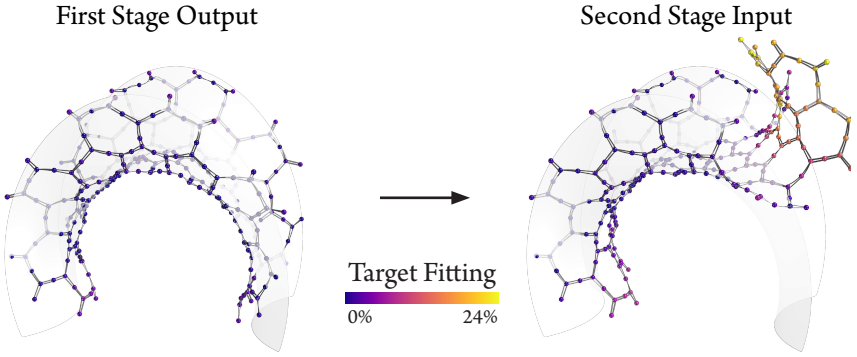


Figure 5.18: A failure case of the relaxation approach in between the two stages of the optimization. Slight perturbations in the assembly rest state may result in irrecoverable buckling configurations.

Fabrication

The simulation frameworks of the three bending-active systems are based on the discrete elastic rod model [Bergou et al., 2008] which has been extensively validated [Romero et al., 2021].

Umbrella Meshes and C-shells fabricate physical prototypes to validate the simulation results. Our rationalization choices to build these systems with a kit-of-parts do not alter the fabrication process and our simulation uses the same frameworks. However, it is necessary to validate the rationalized designs with physical prototypes given the sensitivity of the equilibrium states. We leave this for future work since designing easily reconfigurable systems also comes with significant engineering challenges.

5.10 Conclusion

An optimized kit of parts enables cost-effective and re-use-friendly manufacturing of complex structures. The kit-of-parts approach is particularly attractive for bending-active structures since each part can appear in different configurations i.e., deformed states, of different structures resulting in non-trivial coupling compared to rigid kit-of-parts assemblies. Furthermore, these structures sharing parts are highly sensitive to perturbations of the part geometries due to active bending. Our computational pipeline enables evaluating the trade-offs between the design preservation of input designs and part set size, by leveraging physical simulation of the bending-active equilibria to guide the optimization of the kit of parts.

Note

This chapter is based on the following publication:

> **Quentin Becker***, Uday Kusupati*, Seiichi Suzuki, Mark Pauly (*joint first authors). Computational Design of a Kit of Parts for Bending-Active Structures. *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia 2024)*.

The candidate jointly contributed to the formalization of the problem, its relaxation, its generic implementation, and to the writing. The candidate exclusively implemented all the experiments related to rationalized C-shells and Orthogonal Grids. All the experiments related to Umbrella Meshes were exclusively realized by Uday Kusupati.

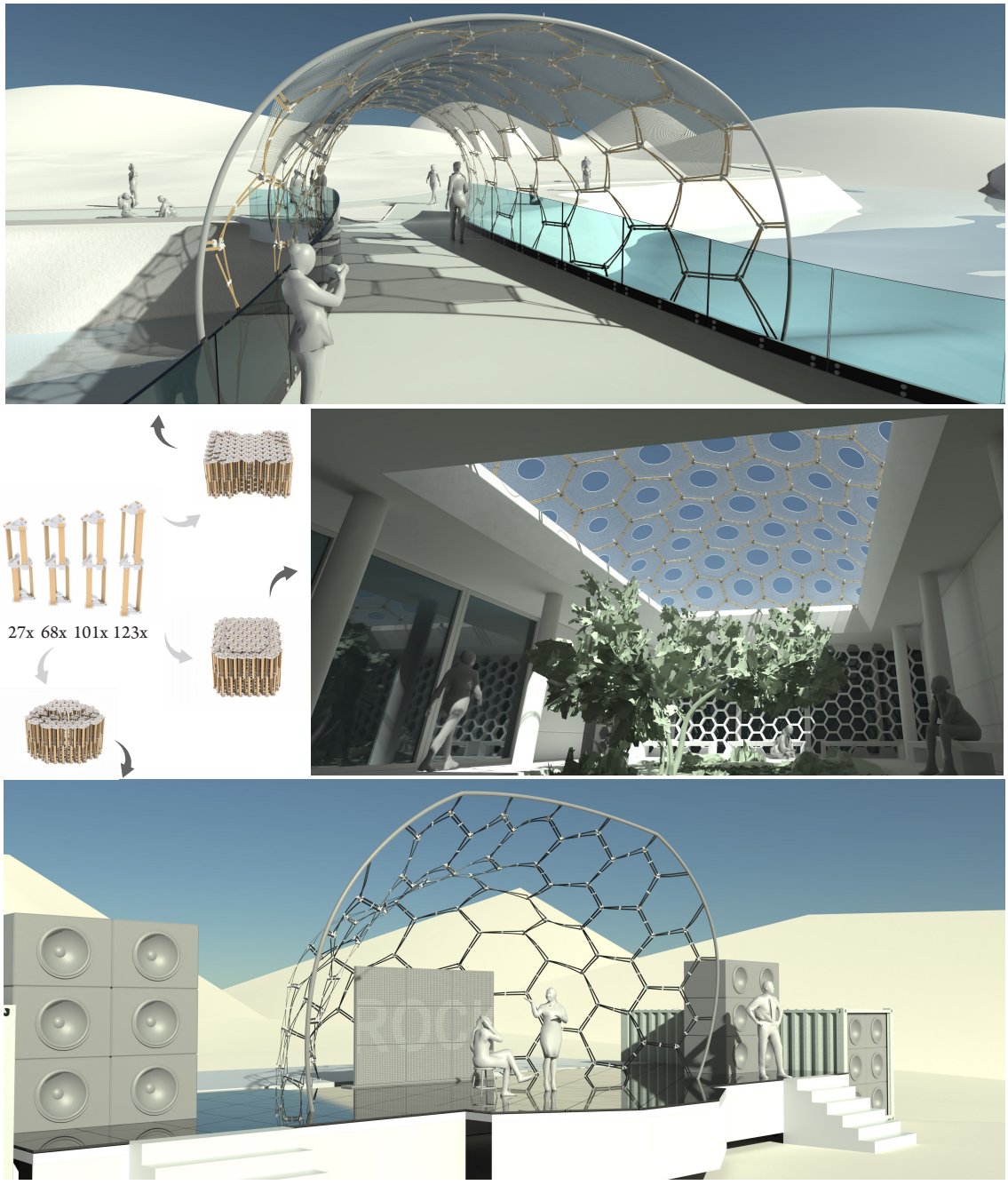


Figure 5.19: Architectural applications: Speculative designs realized with an optimized kit of parts.

Chapter 6

Conclusion

Deployable gridshells have emerged as a lightweight and load-bearing structural typology that can be used for a variety of applications, from temporary shelters to architectural pavilions. Designing these structures, and particularly deployable gridshells is a challenging task, where fabricability constraints thwarts the modeling of complex geometries. This thesis has introduced a new class of gridshells that mitigates assemblability constraints by using curved elastic beams in the linkage: C-shells possess a guaranteed flat and stress-free assembly state. Additionally, C-shells present a remarkable diversity in the range of deployed shapes they can assume.

This work presented a computational forward and inverse design pipeline for C-shells. The forward approach allows exploring the shape space of C-shells through an efficient physics-based simulation. Inverse design relies on a novel geometric flattening algorithm that computes a candidate planar layout of beams that respects kinetic constraints necessary for it to deploy into a user-defined target surface. The resulting linkage can be further improved using our design optimization algorithm.

Efficiency of the forward exploration tool is crucial for understanding the shape space spanned by the deployed states of C-shells. We exposed two key concepts inherent to C-shells deployment i.e., linkage deployability and sensitivity to uncoordinated perturbations of the joints positions. This thesis introduced a set of parametric families of design edits based on conformal maps which satisfies the deployability and sensitivity requirements. We demonstrated the efficiency of our approach in generating a wide variety of designs with different grid topologies. A Rhino-Grasshopper plugin was developed to allow designers and architects to interactively explore the design space of C-shells.

The numerous advantages brought by C-shells come with a cost: The increased complexity of the constitutive elements. The use of curved beams requires more advanced fabrication techniques e.g., through laser cutting or 3D printing, and produce additional material waste compared to straight beams. This thesis tackles the rationalization of curved elements into a sparse kit of parts made of simple linear elements. Rationalization takes the form of a mixed-integer optimization problem, where elements are assigned to parts and parts' geometries are updated to best fit the user-provided designs. A two-stage relaxation of the original combinatorial problem was pro-

posed to make the optimization tractable. We demonstrated the efficiency of our computational design framework on enhancing part reuse on three different classes of bending-active structures, showcasing at the same time the versatility of our approach.

This thesis takes a step towards democratizing the design of deployable gridshells, but, to fully realize the potential of C-shells, several challenges of theoretical, engineering, and artistic nature remain to be explored. We list some of the most promising directions for future work in the following section.

6.1 Future Work

Topology Optimization

In this thesis, we assume the gridshell topology to be user-provided. Figure 3.19 illustrates a shortcoming of this approach in the context of C-shells inverse design: The topology may not be suitable for the target shape. In this example, it is clear that some of the four quadrilaterals at the corners need to be discarded to reduce the distribution of joint opening angles and thus improve the deployability capabilities of the structure. A heuristic-based pruning approach is conceivable, but it is not guaranteed to find a topology adapted to the surface depending on the initial guess.

Rather than a palliative solution, a more robust approach would be to integrate topology exploration into the planarization step or in an upstream step. Deriving a (almost-everywhere) continuous relaxation of the problem e.g., by tracing selected curves on the target surface and intersecting them, could be a promising direction and would render the combinatorial problem tractable.

Diffusion models have recently gained enormous traction in the Machine Learning community, be it for visual computing [Po et al., 2024] or protein generation [Watson et al., 2023]. Equipped with the parametric families presented in this thesis as data sources, a diffusion model could be trained to learn the connection between grid topologies and surfaces. This presupposes an adapted representation of C-shells and target shapes, and an efficient data generation strategy.

String Actuation and Shape Transformation

The deployable gridshells presented in this thesis exploit torque actuation uniformly at the joints to deploy. Even when sparsifying the number of actuators, this method is not practical and induces undesired localized load on the structure. In the line of tensile-actuated deformable objects [Skouras et al., 2013, Dandy et al., 2024], deploying C-shells using a sensibly threaded cable network could be a viable solution. Actuators sparsity would be promoted in the new computational design pipeline.

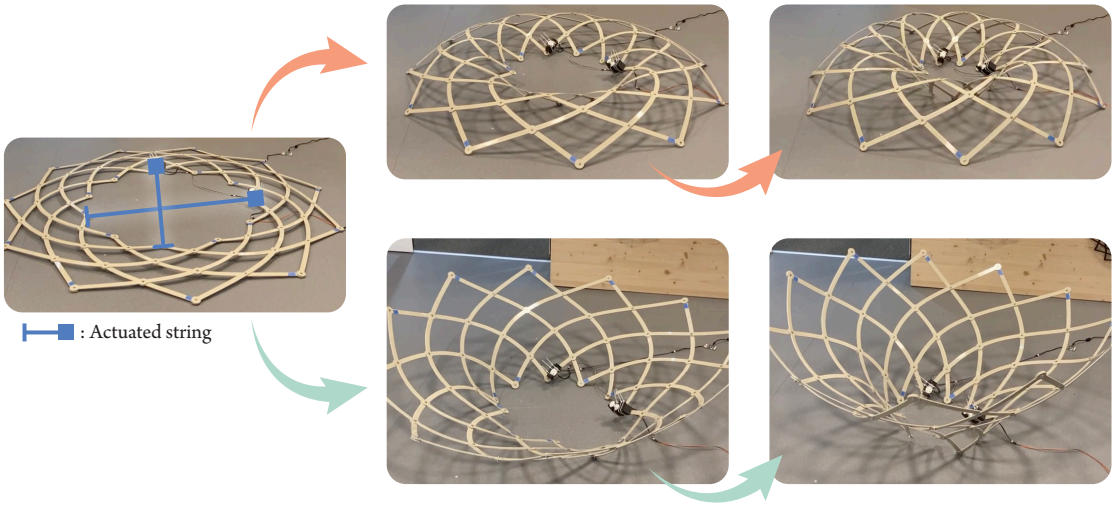


Figure 6.1: A study on the deployment of C-shell using tensile elements. The structure is actuated by winding the cables incrementally. Two different equilibrium states are reached by manually acting on the structure at the beginning of the deployment.

Carefully routing the cables could also serve the purpose of driving the deployment path towards one or the other equilibrium state. The ability to reach distinct states is the starting point for locomotion, which opens a new application domain for deployable gridshells. We were able to demonstrate the applicability of this approach on the multistable (half-)torus C-shell shown in Figure 3.20 during a joint workshop with Valentina Soana's class of architect students in April 2024. Two strings are used to actuate the structure as depicted in Figure 6.1. By reducing the string lengths through winding, the structure can be morphed into two different equilibrium states. For a fully autonomous deployment, an *extrinsic* cable could be added between a joint in the inner ring and the outer ring of the linkage. Extrinsic cables are not constrained to follow the surface approximated by the actuated structure: The two strings in Figure 6.1 are extrinsic. On the other hand, intrinsic cables stay close to the surface during deployment e.g., they connect a succession of joints following quadrilateral diagonals. A manually crafted combination of intrinsic and extrinsic cables have been designed to actuate a demonstrator into an undulating motion [Ono et al., 2024].

More generally, the problem of finding the optimal cable routing strategy, both in terms of sparsity and energy or stress distribution, to reach two or more equilibrium states is an interesting yet challenging problem.

Figure 6.2: Example of elastic structures combined with surface elements. From left to right: The X-shell pavilion by Isvoranu and colleagues 2019, the Solidays Forum by the THINkSHELL group [Baverel et al., 2012], and the el.ba by Alison Martin and the EPFL Geometric Computing Laboratory.

Balancing Gridshells and Elastic Membranes

Deployable gridshells collapse to their rest state when released. In practice, we tighten screws at the joints sufficiently to keep the gridshell in its deployed state. Isvoranu and colleagues 2019 covered the structure with a decorative membrane to give a sense of enclosure. This membrane does not contribute to the structure's stability. The Solidays Forum designed by the THINkSHELL group also employs an elastic membrane to wrap the whole structure. However, no quantitative assessment of the membrane's contribution to the stability of the structure has been provided [Baverel et al., 2012]. Friction between the membrane and the beams plays a major part on that matter and may be challenging to model accurately.

Recently, in the artistic project "el.ba" born from the collaboration between the weaving artist Alison Martin and the EPFL Geometric Computing Laboratory (GCM), elastic rod and membrane communicate and balance the whole structure into a static equilibrium. Additionally, the membrane patches have been designed to distribute the stress as isotropically as possible and prevent visually unpleasant wrinkles.

The computational design framework presented in this thesis could be extended to incorporate a membrane elasticity model in the inner equilibrium solve. However, initializing the design optimization algorithm is contingent upon an understanding of the interplay between rod and surface elements.

Bibliography

Lara Alegria Mira, Ashley P Thrall, and Niels De Temmerman. The universal scissor component: optimization of a reconfigurable component for deployable scissor structures. *Engineering optimization*, 48(2):317–333, 2016.

David Arthur, Sergei Vassilvitskii, et al. k-means++: The advantages of careful seeding. In *Soda*, volume 7, pages 1027–1035, 2007.

Guy Austern, Isaac Guedi Capeluto, and Yasha Jacob Grobman. Rationalization methods in computer aided fabrication: A critical review. *Automation in Construction*, 90:281–293, 2018.

Phil Ayres, Alison Grace Martin, and Mateusz Zwierzycki. 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)*, 2018.

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

Changyeob Baek, Andrew O. Sageman-Furnas, Mohammad K. Jawed, and Pedro M. Reis. Form finding in elastic gridshells. *Proceedings of the National Academy of Sciences*, 115(1):75–80, 2018. doi: 10.1073/pnas.1713841115. URL `https://www.pnas.org/doi/abs/10.1073/pnas.1713841115`.

Changyeob Baek, Alison G. Martin, Samuel Poincloux, Tian Chen, and Pedro M. Reis. Smooth triaxial weaving with naturally curved ribbons. *Phys. Rev. Lett.*, 127:104301, Aug 2021. doi: 10.1103/PhysRevLett.127.104301.

Michael R. Barnes. *Form-finding and Analysis of Tension Space Structures by Dynamic Relaxation*. University of The City, 1977.

Paolo Basso, Andrea E Del Grosso, Alberto Pugnale, and Mario Sassone. Computational morphogenesis in architecture: Cost optimization of free-form grid shells. *Journal of the international association for shell and spatial structures*, 50(3):143–150, 2009.

Olivier Baverel, Jean-François Caron, Frederic Tayeb, and Lionel Du Peloux. Gridshells in composite materials: construction of a 300 m2 forum for the solidays' festival in paris. *Structural engineering international*, 22(3):408–414, 2012.

Quentin Becker, Seiichi Suzuki, Yingying Ren, Davide Pellis, Julian Panetta, and Mark Pauly. C-shells: Deployable gridshells with curved beams. *ACM Trans. Graph.*, 42(6), dec 2023. ISSN 0730-0301. doi: 10.1145/3618366.

Quentin Becker, Seiichi Suzuki, and Mark Pauly. Interactive design of c-shells using reduced parametric families. In *Proceedings of the IASS Annual Symposium*. International Association for Shell and Spatial Structures (IASS), 2024.

Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. Discrete elastic rods. *ACM Transactions on Graphics (TOG)*, 27(3):1–12, 2008.

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

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

Richard L Bishop. There is more than one way to frame a curve. *The American Mathematical Monthly*, 82(3):246–251, 1975. doi: https://doi.org/10.1080/00029890.1975.11993807.

J William Boley, Wim M Van Rees, Charles Lissandrello, Mark N Horenstein, Ryan L Truby, Arda Kotikian, Jennifer A Lewis, and L Mahadevan. Shape-shifting structured lattices via multimaterial 4d printing. *Proceedings of the National Academy of Sciences*, 116(42):20856–20862, 2019.

Giulia Boller, Philippe Block, and Joseph Schwartz. Heinz isler's physical form finding of the hib tennis shells. *Structures*, 65:106559, 2024. ISSN 2352-0124. doi: https://doi.org/10.1016/j.istruc.2024.106559. URL `https://www.sciencedirect.com/science/article/pii/S2352012424007112`.

Jan Brütting, Gennaro Senatore, and Corentin Fivet. Form follows availability–designing structures through reuse. *Journal of the International Association for Shell and Spatial Structures*, 60 (4):257–265, 2019.

Jan Brütting, Gennaro Senatore, and Corentin Fivet. Design and fabrication of a reusable kit of parts for diverse structures. *Automation in Construction*, 125:103614, 2021.

B. Burkhardt and F. Otto. *Multihalle Mannheim; the documentation on the design and execution work of Mannheim Hall*. IL (Collection). Krämer, 1978. ISBN 9783782820134.

Rulin Chen, Pengyun Qiu, Peng Song, Bailin Deng, Ziqi Wang, and Ying He. Masonry shell structures with discrete equivalence classes. *ACM Transactions on Graphics*, 42(4), 2023.

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

Eustathios D Chiotis. Pythagoras'mathematics in architecture and his influence on great cultural works. *Scientific Culture*, 7(1), 2021.

Anthony S Ramos Cisneros, Alisher Aikyn, Martin Kilian, Helmut Pottmann, and Christian Müller. Approximation by meshes with spherical faces. *ACM Trans. Graph.*, 43(6), December 2024. doi: 10.1145/3687942. URL https://doi.org/10.1145/3687942.

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

Lilian-Joy Yana Dandy, Michele Vidulis, Yingying Ren, and Mark Pauly. Tencers: Tension-constrained elastic rods. *ACM Trans. Graph.*, 43(6), dec 2024. doi: 10.1145/3687967.

Cyril Douthe, Romain Mesnil, Hugo Orts, and Olivier Baverel. Isoradial meshes: Covering elastic gridshells with planar facets. *Automation in Construction*, 83:222–236, 2017. ISSN 0926-5805. doi: 10.1016/j.autcon.2017.08.015.

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

Jean Nicolas Louis Durand. *Précis des leçons d'architecture données à l'École polytechnique*, volume 2. chez l'auteur, à l'Ecole polytechnique, 1809.

Steinar Hillersøy Dyvik, Bendik Manum, and Anders Rønnquist. Gridshells in recent research—a systematic mapping study. *Applied Sciences*, 11(24):11731, 2021.

Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 173–182, 1995.

Michael Eigensatz, Mario Deuss, Alexander Schiftner, Martin Kilian, Niloy J Mitra, Helmut Pottmann, and Mark Pauly. Case studies in cost-optimized paneling of architectural freeform surfaces. In *Advances in architectural geometry 2010*, pages 49–72. Springer, 2010a.

Michael Eigensatz, Martin Kilian, Alexander Schiftner, Niloy J. Mitra, Helmut Pottmann, and Mark Pauly. Paneling architectural freeform surfaces. *ACM Trans. Graph.*, 29(4), July 2010b. ISSN 0730-0301. doi: 10.1145/1778765.1778782. URL https://doi.org/10.1145/1778765.1778782.

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

Corentin Fivet and Jan Brütting. Nothing is lost, nothing is created, everything is reused: structural design for a circular economy. *The Structural Engineer*, 98(1):74–81, 2020.

Chi-Wing Fu, Chi-Fu Lai, Ying He, and Daniel Cohen-Or. K-set tilable surfaces. *ACM transactions on graphics (TOG)*, 29(4):1–6, 2010.

Akash Garg, Andrew O. Sageman-Furnas, Bailin Deng, Yonghao Yue, Eitan Grinspun, Mark Pauly, and Max Wardetzky. Wire mesh design. *ACM Trans. Graph.*, 33(4), jul 2014. ISSN 0730-0301. doi: 10.1145/2601097.2601106.

Grégoire Gaudreault and Andrei Nejur. Heteromorph: Temporary shelters made of reclaimed heterogeneous materials. In *Proceedings of eCAADe*, pages 59–68, 09 2023. doi: 10.52842/conf.ecaade.2023.2.059.

Konstantinos Gavriil, Ruslan Guseinov, Jesús Pérez, Davide Pellis, Paul Henderson, Florian Rist, Helmut Pottmann, and Bernd Bickel. Computational design of cold bent glass façades. *ACM Trans. Graph.*, 39(6), nov 2020. ISSN 0730-0301. doi: 10.1145/3414685.3417843. URL `https://doi.org/10.1145/3414685.3417843`.

Mark Gorgolewski. *Resource salvation: the architecture of reuse*. John Wiley & Sons, 2017.

Ruslan Guseinov, Eder Miguel, and Bernd Bickel. Curveups: Shaping objects from flat plates with tension-actuated curvature. *ACM Trans. Graph.*, 36(4), 7 2017. doi: 10.1145/3072959.3073709.

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

Christian Hafner and Bernd Bickel. The design space of kirchhoff rods. *ACM Trans. Graph.*, 42 (5), September 2023. ISSN 0730-0301. doi: 10.1145/3606033. URL `https://doi.org/10.1145/3606033`.

Charles A Hall and W Weston Meyer. Optimal error bounds for cubic spline interpolation. *Journal of Approximation Theory*, 16(2):105–122, 1976.

Edmund Happold and LIDDELL WI. Timber lattice roof for the mannheim bundesgartenschau. *The Structural Engineer*, 53(3), 1975.

Richard Harris and Jonathan Roynon. The savill garden gridshell design and construction. In *10th World Conference of Timber Engineering*, 2008.

Richard Harris, John Romer, Oliver Kelly, and Stephen Johnson. Design and construction of the downland gridshell. *Building Research & Information*, 31(6):427–454, 2003.

A Scott Howe, Isamu Ishii, and Tomohiro Yoshida. Kit-of-parts: A review of object-oriented construction techniques. In *ISARC'99: international symposium on automation and robotics in construction (Madrid, 22-24 September 1999)*, pages 165–171, 1999.

Yijiang Huang, Latifa Alkhayat, Catherine De Wolf, and Caitlin Mueller. Algorithmic circular design with reused structural elements: method and tool. In *Proceedings of the International fib Symposium on the Conceptual Design of Structures*, volume 55, pages 457–468. Fédération Internationale du Béton (fib), 2021.

Florin Isvoranu, Julian Panetta, Tian Chen, Etienne Bouleau, and Mark Pauly. X-shell pavilion: A deployable elatic rod structure. *Proceedings of the IASS Annual Symposium - Structural Membranes 2019*, 2019.

Caigui Jiang, Jun Wang, Johannes Wallner, and Helmut Pottmann. Freeform honeycomb structures. *Computer Graphics Forum*, 33(5):185–194, 2014.

Caigui Jiang, Hui Wang, Victor Ceballos Inza, Felix Dellinger, Florian Rist, Johannes Wallner, and Helmut Pottmann. Using isometries for computational design and fabrication. *ACM Transactions on Graphics (TOG)*, 40(4):1–12, 2021.

David Jourdan, Melina Skouras, Etienne Vouga, and Adrien Bousseau. Computational design of self-actuated surfaces by printing plastic ribbons on stretched fabric. *Computer Graphics Forum*, 41(2), April 2022. URL `http://www-sop.inria.fr/reves/Basilic/2022/JSVB22`.

Martin Kilian, Anthony S Ramos Cisneros, Christian Müller, and Helmut Pottmann. Meshes with spherical faces. *ACM Trans. Graph.*, 42(6), December 2023. ISSN 0730-0301. doi: 10.1145/3618345. URL `https://doi.org/10.1145/3618345`.

Joseph C Klann. Walking device, April 2 2002. US Patent 6,364,040.

Jan Knippers, Jan Cremers, Markus Gabler, and Julian Lienhard. *Construction Manual for Polymers and Membranes*. 10 2011. ISBN 978-3-0346-0726-1.

Jan Knippers, Florian Scheible, Matthias Oppe, and Hauke Jungjohann. Bio-inspired kinetic gfrp-façade for the thematic pavilion of the expo 2012 in yeosu. In *International Symposium of Shell and Spatial Structures (IASS 2012)*, volume 90, pages 341–347, 2012.

Mina Konaković, Keenan Crane, Bailin Deng, Sofien Bouaziz, Daniel Piker, and Mark Pauly. Beyond developable: computational design and fabrication with auxetic materials. *ACM Transactions on Graphics (TOG)*, 35(4):1–11, 2016.

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

Uday Kusupati, Florin Isvoranu, Seiichi Suzuki, and Mark Pauly. Rum: Reconfigurable umbrella mesh. *Advances in Architectural Geometry 2023*, page 474, 2023.

Riccardo La Magna. *Bending-Active Plates: Strategies for the Induction of Curvature through the Means of Elastic Bending of Plate-based Structures*. Forschungsberichte aus dem Institut für Tragkonstruktionen und konstruktives Entwerfen der Universität Stuttgart. Universität Stuttgart Inst. f. Tragkonstr., 2017. ISBN 9783922302438.

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

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

Lev D Landau, Evgenıı M Lifshitz, RJ Atkin, and N Fox. The theory of elasticity. In *Physics of Continuous Media*, pages 167–178. CRC Press, 2020.

Lev Davidovich Landau, Arnold Markovich Kosevich, Evgenii Mikhailovich Lifshitz, and Lev Petrovich Pitaevskii. *Theory of Elasticity: Volume 7*. Course of theoretical physics. Elsevier Science, 1986. ISBN 9780750626330. URL https://books.google.com/books?id=tpY-VkwCkAIC.

Julian Lienhard. *Bending-Active Structures: Form-finding Strategies Using Elastic Deformation in Static and Kinetic Systems and the Structural Potentials Therein*. Forschungsberichte aus dem Institut für Tragkonstruktionen und konstruktives Entwerfen der Universität Stuttgart. Universität Stuttgart Inst. f. Tragkonstr., 2014. ISBN 9783922302360.

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

Julian Lienhard, Holger Alpermann, Christoph Gengnagel, and Jan Knippers. Active bending, a review on structures where bending is used as a self-formation process. *International Journal of Space Structures*, 28(3-4):187–196, 2013. doi: 10.1260/0266-3511.28.3-4.187. URL https://doi.org/10.1260/0266-3511.28.3-4.187.

Bosheng Liu and Tanvir R. Faisal. Computational design and fabrication of a bending-active structure using fiberglass: A bioinspired pavilion mimicking marine microorganism radiolaria. *Journal of Bionic Engineering*, 19(2), 2022.

Daoming Liu, Davide Pellis, Yu-Chou Chiang, Florian Rist, Johannes Wallner, and Helmut Pottmann. Deployable strip structures. *ACM Trans. Graph.*, 42(4), jul 2023a. ISSN 0730-0301. doi: 10.1145/3592393.

Yang Liu, Helmut Pottmann, Johannes Wallner, Yong-Liang Yang, and Wenping Wang. Geometric modeling with conical meshes and developable surfaces. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933646. doi: 10.1145/1179352.1141941. URL https://doi.org/10.1145/1179352.1141941.

Yizhe Liu, Fei Pan, Bin Ding, Yilong Zhu, Kuijian Yang, and Yuli Chen. Multistable shape-reconfigurable metawire in 3d space. *Extreme Mechanics Letters*, 50:101535, 2022.

Yuanpeng Liu, Ting-Uei Lee, Antiopi Koronaki, Nico Pietroni, and Yi Min Xie. Reducing the number of different nodes in space frame structures through clustering and optimization. *Engineering Structures*, 284:116016, 2023b.

Zhong-Yuan Liu, Zhan Zhang, Di Zhang, Chunyang Ye, Ligang Liu, and Xiao-Ming Fu. Modeling and fabrication with specified discrete equivalence classes. *ACM Trans. Graph.*, 40(4), jul 2021. ISSN 0730-0301. doi: 10.1145/3450626.3459843. URL https://doi.org/10.1145/3450626.3459843.

Hongjia Lu and Yi Min Xie. Reducing the number of different members in truss layout optimization. *Structural and Multidisciplinary Optimization*, 66(3):52, 2023.

Sheng-Jie Luo, Yonghao Yue, Chun-Kai Huang, Yu-Huan Chung, Sei Imai, Tomoyuki Nishita, and Bing-Yu Chen. Legolization: Optimizing lego designs. *ACM Transactions on Graphics (TOG)*, 34(6):1–12, 2015.

James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

Luigi Malomo, Jesús Pérez, Emmanuel Iarussi, Nico Pietroni, Eder Miguel, Paolo Cignoni, and Bernd Bickel. Flexmaps: Computational design of flat flexible shells for shaping 3d objects. *ACM Trans. on Graphics - Siggraph Asia 2018*, 37(6):14, nov 2018. URL http://vcg.isti.cnr.it/Publications/2018/MPIPMCB18. https://doi.org/10.1145/3272127.3275076.

Iason Manolas, Francesco Laccone, Gianmarco Cherchi, Luigi Malomo, Paolo Cignoni, et al. A computational tool for the analysis of 3d bending-active structures based on the dynamic relaxation method. In *STAG*, pages 1–9, 2022.

Robert McNeel et al. Rhinoceros 3d (version 8.0), 2009.

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

Romain Mesnil, Takara Muto, Krittika Walia, Cyril Douthe, and Olivier Baverel. *Design and Fabrication of a Pseudo-Geodesic Gridshell*, pages 83–96. De Gruyter, Berlin, Boston, 2023. ISBN 9783111162683. doi: doi:10.1515/9783111162683-007. URL https://doi.org/10.1515/9783111162683-007.

Saurabh Mhatre, Elisa Boatti, David Melancon, Ahmad Zareei, Maxime Dupont, Martin Bechthold, and Katia Bertoldi. Deployable structures based on buckling of curved beams upon a rotational input (adv. funct. mater. 35/2021). *Advanced Functional Materials*, 31 (35):2170261, 2021. doi: https://doi.org/10.1002/adfm.202170261. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/adfm.202170261.

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

Fuki Ono, Haruto Kamijo, Miwako Kase, Seri Nishimoto, Kotaro Sempuku, Mizuki Shigematsu, and Tomohiro Tachi. Growth-induced transformable surfaces realized by bending-active scissors grid. *Architectural Intelligence*, 3(1):21, 2024.

Frei Otto and Bodo Rasch. *Finding Form*. Axel Menges, 1995. ISBN 9783930698660.

Julian Panetta, Mina Konaković-Luković, Florin Isvoranu, Etienne Bouleau, and Mark Pauly. X-shells: A new class of deployable beam structures. *ACM Transactions on Graphics (TOG)*, 38 (4):1–15, 2019. doi: 10.1145/3306346.3323040.

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

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

Davide Pellis, Martin Kilian, Helmut Pottmann, and Mark Pauly. Computational design of weingarten surfaces. *ACM Trans. Graph.*, 40(4), jul 2021. ISSN 0730-0301. doi: 10.1145/3450626.3459939. URL https://doi.org/10.1145/3450626.3459939.

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

Stefan Pillwein and Przemyslaw Musialski. Generalized deployable elastic geodesic grids. *ACM Trans. Graph.*, 40(6), dec 2021. ISSN 0730-0301. doi: 10.1145/3478513.3480516. URL https://doi.org/10.1145/3478513.3480516.

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

Stefan Pillwein, Johanna Kübert, Florian Rist, and Przemyslaw Musialski. Design and fabrication of multi-patch elastic geodesic grid structures. *Computers and Graphics*, 98:218–230, 2021. ISSN 0097-8493. doi: https://doi.org/10.1016/j.cag.2021.06.002. URL https://www.sciencedirect.com/science/article/pii/S0097849321001163.

R. Po, W. Yifan, V. Golyanik, K. Aberman, J. T. Barron, A. Bermano, E. Chan, T. Dekel, A. Holynski, A. Kanazawa, C.K. Liu, L. Liu, B. Mildenhall, M. Nießner, B. Ommer, C. Theobalt,

P. Wonka, and G. Wetzstein. State of the art on diffusion models for visual computing. *Computer Graphics Forum*, 43(2):e15063, 2024. doi: https://doi.org/10.1111/cgf.15063. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.15063.

Helmut Pottmann, Andreas Asperl, and Axel Kililan. *Architectural geometry*. SIAM, 2007a.

Helmut Pottmann, Yang Liu, Johannes Wallner, Alexander Bobenko, and Wenping Wang. Geometry of multi-layer freeform structures for architecture. In *ACM SIGGRAPH 2007 Papers*, SIGGRAPH '07, New York, NY, USA, 2007b. Association for Computing Machinery. ISBN 9781450378369. doi: 10.1145/1275808.1276458. URL https://doi.org/10.1145/1275808.1276458.

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

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

Yingying Ren, Uday Kusupati, Julian Panetta, Florin Isvoranu, Davide Pellis, Tian Chen, and Mark Pauly. Umbrella meshes: Elastic mechanisms for freeform shape deployment. *ACM Trans. Graph.*, 41(4), 2022.

Victor Romero, Mickaël Ly, Abdullah Haroon Rasheed, Raphaël Charrondière, Arnaud Lazarus, Sébastien Neukirch, and Florence Bertails-Descoubes. Physical validation of simulators in computer graphics: A new framework dedicated to slender elastic structures and frictional contact. *ACM Transactions on Graphics (TOG)*, 40(4):1–19, 2021.

Andrew O. Sageman-Furnas, Albert Chern, Mirela Ben-Chen, and Amir Vaxman. Chebyshev nets from commuting polyvector fields. *ACM Trans. Graph.*, 38(6), nov 2019. ISSN 0730-0301. doi: 10.1145/3355089.3356564. URL https://doi.org/10.1145/3355089.3356564.

Rohan Sawhney and Keenan Crane. Boundary first flattening. *ACM Transactions on Graphics (ToG)*, 37(1):1–14, 2017. doi: 10.1145/3132705.

H.-J. Schek. The force density method for form finding and computation of general networks. *Computer Methods in Applied Mechanics and Engineering*, 3(1):115–134, 1974. ISSN 0045-7825. doi: https://doi.org/10.1016/0045-7825(74)90045-0.

Jonas Schikore and Eike Schling. 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)*, 2021.

Jonas Schikore, Eike Schling, Thomas Oberbichler, and Anna Bauer. Kinetics and design of semi-compliant grid mechanisms. In *Advances in Architectural Geometry (AAG)*, 04 2021.

Eike Schling. *Repetitive Structures*. PhD thesis, Technische Universität München, 2018. URL https://mediatum.ub.tum.de/1449869.

Eike Schling and Rainer Barthel. Repetitive structures. In *Impact: Design With All Senses: Proceedings of the Design Modelling Symposium, Berlin 2019*, pages 360–375. Springer, 2020.

Eike Schling, Martin Kilian, Wang Hui, Jonas Schikore, and Helmut Pottmann. Design and construction of curved support structures with repetitive parameters. In *Advances in Architectural Geometry (AAG)*, 09 2018. ISBN 978-3-903015-13-5.

Eike Schling, Hui Wang, Sebastian Hoyer, and Helmut Pottmann. Designing asymptotic geodesic hybrid gridshells. *Computer-Aided Design*, 152:103378, 2022.

Dennis Robert Shelden. *Digital surface representation and the constructibility of Gehry's architecture*. PhD thesis, Massachusetts Institute of Technology, 2002.

Mayank Singh and Scott Schaefer. Triangle surfaces with discrete equivalence classes. *ACM Transactions on Graphics (TOG)*, 29(4):1–7, 2010.

Mélina Skouras, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, and Markus Gross. Computational design of actuated deformable characters. *ACM Trans. Graph.*, 32(4), July 2013. ISSN 0730-0301. doi: 10.1145/2461912.2461979. URL https://doi.org/10.1145/2461912.2461979.

E. Soriano, R. Sastre, and D. Boixader. 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*, page 1894. International Centre for Numerical Methods in Engineering (CIMNE), 10 2019. URL http://hdl.handle.net/2117/330614.

Caroline St-Hilaire and Andrei Nejur. Woodn - in search of a constructive system for a sustainable temporary architecture. In *Proceedings of eCAADe*, pages 185–194, 01 2022. doi: 10.52842/conf.ecaade.2022.1.185.

Seiichi Suzuki, Alison Martin, Yingying Ren, Tzu-Ying Chen, Stefana Parascho, and Mark Pauly. *BamX: Rethinking Deployability in Architecture through Weaving*. De Gruyter, oct 2023. ISBN 9783111160115. doi: 10.1515/9783111162683-016. URL https://infoscience.epfl.ch/handle/20.500.14299/201580.

Xavier Tellier. Bundling elastic gridshells with alignable nets. part i: Analytical approach. *Automation in Construction*, 141:104291, 2022.

Romain Pierre Testuz, Yuliy Schwartzburg, and Mark Pauly. Automatic generation of constructable brick sculptures. *Eurographics 2013-Short Papers*, pages 81–84, 2013.

Josh Vekhter, Jiacheng Zhuo, Luisa F Gil Fandino, Qixing Huang, and Etienne Vouga. Weaving geodesic foliations. *ACM Trans. Graph.*, 38(4), jul 2019. ISSN 0730-0301. doi: 10.1145/3306346.3323043. URL https://doi.org/10.1145/3306346.3323043.

Robert Venturi. *Complexity and contradiction in architecture*, volume 1. The Museum of modern art, 1977.

Michele Vidulis, Yingying Ren, Julian Panetta, Eitan Grinspun, and Mark Pauly. Computational exploration of multistable elastic knots. *ACM Trans. Graph.*, 42(4), July 2023. ISSN 0730-0301. doi: 10.1145/3592399. URL https://doi.org/10.1145/3592399.

Joseph L Walsh. History of the riemann mapping theorem. *The American Mathematical Monthly*, 80(3):270–276, 1973.

Richard A Waltz and Jorge Nocedal. Knitro 2.0 user's manual. *Ziena Optimization, Inc.*, 7:33–34, 2004.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Xudong Yang, Yuan Zhou, Huichan Zhao, Weicheng Huang, Yifan Wang, K. Jimmy Hsia, and Mingchao Liu. Morphing matter: from mechanical principles to robotic applications. *Soft Science*, 3(4), 2023. ISSN 2769-5441. doi: 10.20517/ss.2023.42. URL https://www.oaepublish.com/articles/ss.2023.42.

Yinan Zhang and Devin Balkcom. Interlocking structure assembly with voxels. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2173–2180. IEEE, 2016.

Tianyu Zhu, Zeng-Hao Xu, Ligang Liu, and Xiao-Ming Fu. Modeling with discrete equivalence classes of planar quads. *Computers & Graphics*, 115:404–411, 2023.

Henrik Zimmer, Marcel Campen, David Bommes, and Leif Kobbelt. Rationalization of triangle-based point-folding structures. *Computer Graphics Forum*, 31(2pt3):611–620, 2012.

Henrik Zimmer, Florent Lafarge, Pierre Alliez, and Leif Kobbelt. Zometool shape approximation. *Graphical Models*, 76(5):390–401, 2014.

Appendix A

Supplementary Material: C-shells

This document provides additional technical details regarding our strategy for enforcing the average opening angle constraint during simulation, our energy non-dimensionalization and stress calculation, our arc-length reparameterization of cubic splines, the terms used our planarization algorithm, the fairness regularization term used in the design optimization objective, and our process for calculating gradients for the design optimization.

A.1 Average Angle Constraint Enforcement

As discussed in Section 4.1 of the main paper, the approach that Panetta et al. [2019] use for imposing the equality constraint on the average opening angles during the deployment simulation is problematic in situations where the deployment path constitutes a direction of negative curvature; in these cases, the elastic energy Hessian H they attempt to factorize at each Newton step is indefinite even at minimizers of the constrained optimization problem. As a result, the solver applies unnecessary modifications to the Hessian (severely damping steps and slowing progress) and refuses to converge, falsely believing the minimum to be a saddle point.

We address this issue by applying a sparse linear change of variables to make the average angle (i.e., the deployment path parameter) an explicit variable of the optimization.

Change of Simulation Variables

A naïve choice for this change of variables can easily make the transformed Hessian dense. For instance, solving for the last joint opening angle variable in terms of the average angle and the other $n-1$ opening angles corresponds to the change of variables:

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{n_{\text{joints}}-2} \\ \alpha_{n_{\text{joints}}-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ -1 & -1 & -1 & \dots & -1 & n \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{n_{\text{joints}}-2} \\ \bar{\alpha}_{\mathbf{x}} \end{bmatrix},$$

the last dense row of which would contribute a dense $n \times n$ block to the transformed Hessian. However, other choices are possible: the important property is that the left $n-1$ columns must average to zero (i.e., be orthogonal to the constraint and not influence the average deployment angle), while the last column must average to one.

We propose a different symmetric formula that achieves this goal while minimizing the number of nonzero entries in any row, thereby *minimizing the fill-in of the transformed Hessian:*

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{n_{\text{joints}}-2} \\ \alpha_{n_{\text{joints}}-1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 1 \\ -1 & 1 & 0 & \dots & 0 & 1 \\ 0 & -1 & 1 & \dots & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 1 \\ 0 & 0 & 0 & \dots & -1 & 1 \end{bmatrix}}_{A_\alpha} \begin{bmatrix} \bar{\alpha}_0 \\ \bar{\alpha}_1 \\ \bar{\alpha}_2 \\ \vdots \\ \bar{\alpha}_{n_{\text{joints}}-2} \\ \bar{\alpha}_{\mathbf{x}} \end{bmatrix}.$$

The new joint angle variables α_0 now correspond to simultaneously opening and closing a pair of joints at unit speed (preserving the average angle).

Here A_α is the joint-angle sub-block of the full sparse matrix A that implements the change of variables $\mathbf{x} = A\bar{\mathbf{x}}$ (expressing the original simulation variables $\mathbf{x}$ in terms of the new variables $\bar{\mathbf{x}}$). The rest of the rows/columns of A are taken from the $n_{\mathbf{p}} \times n_{\mathbf{p}}$ identity matrix so that only the joint angles are affected by the change of variables.

Our sparse change of variables can be adapted for the case that only a subset I of the joints are actuated. We then construct a smaller $|I| \times |I|$ matrix A_α using the same approach above but restricted to the angle variables of actuated joints. The remaining rows/columns of A are once more taken from the identity matrix.

We use these new simulation variables $\bar{\mathbf{x}}$ throughout our pipeline, but refer to them as $\mathbf{x}$ in the main paper to avoid cluttered notation.

Impact on the Gradients and Hessians

The gradient of the total elastic energy $\frac{\partial}{\partial \bar{\mathbf{x}}} E(A\bar{\mathbf{x}}, \mathbf{p})$ can be obtained via the chain rule as $\frac{\partial E}{\partial \mathbf{x}} A$. Similarly, the Hessian of the total elastic energy $\frac{\partial^2}{\partial \bar{\mathbf{x}}^2} E(A\bar{\mathbf{x}}, \mathbf{p})$ is given by

$$\frac{\partial}{\partial \bar{\mathbf{x}}} \left(\frac{\partial E}{\partial \mathbf{x}} A \right) = A^\mathsf{T} \frac{\partial^2 E}{\partial \mathbf{x}^2} A.$$

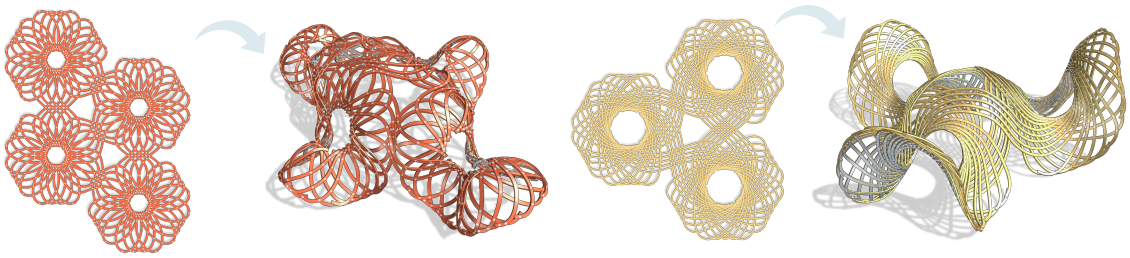


Figure A.1: Some patterns can be repeated and stitched in the flat layout during the design conception phase to generate interesting deployed states.

To study the impact of this change of variables on the Hessian sparsity pattern, we note that

$$A^{\mathsf{T}} \frac{\partial^2 E}{\partial \mathbf{x}^2} A = \sum_{k,l} h_{k,l} \mathbf{a}_k^{\top} \mathbf{a}_l,$$

where $h_{k,l}$ is an element of the original Hessian $\frac{\partial^2 E}{\partial \mathbf{x}^2}$, and *row vector* $\mathbf{a}_k$ denotes the k^{th} row of A, making the expression $\mathbf{a}_k^{\top} \mathbf{a}_l$ an outer product. Since these rows all contain a 1 in the location associated with average angle variable $\bar{\alpha}_{\mathbf{x}}$, the sum of outer products will fill in the row and column of $\frac{\partial^2 E}{\partial \bar{\mathbf{x}}^2}$ associated with $\bar{\alpha}_{\mathbf{x}}$. However, this variable is pinned during deployment, *removing the highly dense row and column of the transformed Hessian*. After this removal, our change of variables leaves every row/column of the transformed Hessian highly sparse. In total, it introduces $O(n_{\text{joints}})$ new nonzero entries, far fewer than the $O(n_{\text{joints}}^2)$ entries introduced by the naïve approach.

Impact on the Deployment

In [Panetta et al., 2019], the deployed equilibrium state for a given opening angle $\bar{\alpha}^{\text{tgt}}$ was found by solving a constrained optimization problem

$$\begin{aligned} \mathbf{x}_{3\text{D}}^*(\mathbf{p}, \bar{\alpha}^{\text{tgt}}) := \underset{\mathbf{x}}{\operatorname{argmin}}\ & E(\mathbf{x}, \mathbf{p}) \\ \text{s.t.} \quad & \mathbf{a}^{\mathsf{T}} \mathbf{x} = \bar{\alpha}^{\text{tgt}}, \end{aligned}$$

where the inner product with vector $\mathbf{a}$ extracts and averages the X-shell's opening angle variables. The associated KKT system,

$$\begin{bmatrix} \dfrac{\partial^2 E}{\partial \mathbf{x}^2} & \mathbf{a} \\ \mathbf{a}^{\mathsf{T}} & 0 \end{bmatrix} \begin{bmatrix} \delta \mathbf{x} \\ \delta \tilde{\lambda} \end{bmatrix} = \begin{bmatrix} -\left(\dfrac{\partial E}{\partial \mathbf{x}} \right)^{\mathsf{T}} \\ \delta \bar{\alpha} \end{bmatrix},$$

determines the infinitesimal change $\delta \mathbf{x}$ in state variable $\mathbf{x}_{3\text{D}}^*(\bar{\mathbf{p}})$ induced by incrementing the average opening angle by $\delta \bar{\alpha}$, where $\delta \tilde{\lambda}$ is the Lagrange multiplier associated to the minimization

problem. The Hessian of the energy evaluated at the current intermediate deployed state $\mathbf{x}^*_{3\mathrm{D}}(\bar{\mathbf{p}})$ may not be positive definite. Formerly, the Hessian would be modified in this case to be positive definite.

After our change of variables, the KKT system becomes

$$\begin{cases} \dfrac{\partial^2 E}{\partial \bar{\mathbf{x}}_{\setminus\bar{\alpha}}{}^2}\delta\bar{\mathbf{x}}_{\setminus\bar{\alpha}} & = -\left(\dfrac{\partial E}{\partial \bar{\mathbf{x}}_{\setminus\bar{\alpha}}}\right)^{\mathsf{T}} \\ \delta\bar{\alpha}_{\mathbf{x}} & = \delta\bar{\alpha}, \end{cases}$$

where we dropped the row and column of the left-hand side of the KKT system that are associated to the pinned average angle variable. The transformed Hessian of the total elastic energy $\frac{\partial^2 E}{\partial \bar{\mathbf{x}}_{\setminus\bar{\alpha}}{}^2}$ is now ensured to be positive definite at the intermediate deployed equilibrium state $\bar{\mathbf{x}}^*_{3\mathrm{D}}(\bar{\mathbf{p}})$. This allows factorizing the left-hand side using an efficient sparse Cholesky factorization.

The total torque applied at the actuated joints to hold the deployed configuration can now be obtained by looking at the component of the gradient associated to the average opening angle

$$\sum_{i=0}^{n_{\text{joints}}-1} \frac{\partial E}{\partial \alpha_i} = \sum_{i=0}^{n_{\text{joints}}-1} \frac{\partial E}{\partial \bar{\alpha}_{\mathbf{x}}}\frac{\partial \bar{\alpha}_{\mathbf{x}}}{\partial \alpha_i} = \frac{\partial E}{\partial \bar{\alpha}_{\mathbf{x}}},$$

where all the gradients are evaluated at $\bar{\mathbf{x}}^*_{3\mathrm{D}}$, the deployed state.

A.2 Non-dimensionalization of the Energy During Deployment

Using the new parameterization, the deployed state is given by

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

where E is now the total elastic energy of the system using the new parameterization, T is the non-dimensionalized target attraction term, $\bar{\alpha}(\bar{\mathbf{x}})$ extracts the average angle variable from $\bar{\mathbf{x}}$, and $\bar{\alpha}^{\mathrm{tgt}}$ is the target average opening angle in the deployed state. The target attraction term is used here only to factor out rigid motion and is scaled by a small weight. To ease parameter tuning, we non-dimensionalize the energy term by dividing it by both the Young's modulus Y of the fabrication material and the rest volume V_0 of the C-shell. This normalization factor is motivated by the fact that the bending, twisting, and stretching energies stored in the rods of a C-shell can be

derived by plugging strain tensor fields $\boldsymbol{\epsilon}$ induced these deformation modes into the linear elasticity energy $\frac{1}{2}\int_\Omega \boldsymbol{\epsilon} : \mathbf{C} : \boldsymbol{\epsilon}$, where C is the fabrication material's elasticity tensor. For an isotropic material, $\mathbf{C} = Y\mathbf{C_0}(\nu)$, where $\mathbf{C_0}(\nu)$ depends only on the Poisson's ratio ν; for example $\mathbf{C_0}(0)$ is the fourth-order identity tensor. Since strain $\boldsymbol{\epsilon}$ is non-dimensional, this elastic energy clearly is proportional to YV_0.

A.3 Stress Computation

We use a similar approach to [Megaro et al., 2017] to compute stresses for Discrete Elastic Rods (DERs). Introducing a 2D coordinate system (x, y) for the material cross-section with the centerline at its origin, we seek to approximate the stress tensor $\boldsymbol{\sigma}(x, y)$. Following the kinetic assumptions of Kirchhoff rods, the state of stress in a deformed rod with torsion τ, *material curvature vector* $\boldsymbol{\kappa}$, and uniaxial stretching $\frac{\Delta l}{l}$ is:

$$\boldsymbol{\sigma}(x,y) = \tau\mu \begin{bmatrix} 0 & 0 & \dfrac{\partial\psi}{\partial x} - y \\ 0 & 0 & \dfrac{\partial\psi}{\partial y} + x \\ \dfrac{\partial\psi}{\partial x} - y & \dfrac{\partial\psi}{\partial y} + x & \dfrac{\sigma_{zz}}{\tau\mu} \end{bmatrix}, \qquad \sigma_{zz} := Y\left(\frac{\Delta l}{l} + \boldsymbol{\kappa}\cdot\begin{bmatrix} x \\ y \end{bmatrix}\right),$$

where the scalar field $\psi(x, y)$ describes how the initially planar cross-section warps out of plane under torsion to relax into equilibrium; it is determined by solving a Laplace equation [Landau et al., 1986, Megaro et al., 2017, Panetta et al., 2019].

We use von Mises stress $\sigma_v = \sqrt{\frac{3}{2}}\left\|\boldsymbol{\sigma}_d\right\|_F$ to evaluate our structures' robustness, where

$$\boldsymbol{\sigma}_d := \boldsymbol{\sigma} - \frac{1}{3}\mathrm{tr}(\boldsymbol{\sigma})\mathbf{I}$$

is the deviatoric stress. This von Mises stress simplifies to:

$$\sigma_v^2 = 3(\mu\tau)^2 \left\| \nabla\psi + \begin{bmatrix} -y \\ x \end{bmatrix} \right\|^2 + \sigma_{zz}^2,$$

and it can be shown that each of these two terms is subharmonic. This means σ_v^2 itself is subharmonic and therefore σ_v satisfies a maximum principle: the maximum von Mises stress occurs on the boundary. In other words, to evaluate failure likelihood, we can restrict our consideration to the cross-section boundary.

One notable departure of our formulation from Megaro et al. [2017] is in its discretization for DERs. We recall that the material curvature $\boldsymbol{\kappa}$ is defined by decomposing the curvature normal into its components with respect to the orthonormal material frame. Unfortunately, for DERs, discrete curvature is defined at the vertices, while the material frames are attached to edges. Megaro et al. [2017] resolve this misalignment by simply averaging the material frame vectors onto the vertices using an arithmetic mean, which is nonphysical. Instead, when evaluating stress at a sample point within the cross-section at a rod vertex, we compute two separate stress values: one according to each incident edge's material frame. Then we average these two values using the incident edge lengths as weights to obtain a single stress sample. We sample stress at every boundary node of the same finite element mesh of the cross-section that we use to solve for ψ, guaranteeing that the maximum is sampled.

A.4 Curve Parameterization

The rest shapes of the beams composing C-shells are parameterized using interpolating cubic splines. For joints and intermediate control points along a curve, we fit a natural cubic spline to join them. The interpolation property is important for preserving the topology of the linkage (so that curve intersections occur precisely at the joints where they are intended).

Curve Discretization

Our curves are split into segments at the joints, and each segment is discretized into edges. The joint model we use requires the segments to overlap at the joints, and share an edge. That model distributes the total rest length of a segment evenly across the edges except the boundary edges. Such shared edges have a length equal to the minimum length of edges from either one of the two consecutive segments.

In order for the discretized linkage to have an energy-free rest state, the rest curvatures measured between two consecutive edges should be carefully picked. Discretizing the curves requires a special care regarding the edge lengths, which we handle by using a arc-length parameterization of our splines. We describe our reparameterization method in the next section. Discrete points can be sampled along the curve so that they respect the above rules on the rest length distribution. Afterwards, we calculate the discrete rest curvature from the turning angle between consecutive edges as explained in [Bergou et al., 2008, 2010].

This entire process, going from the design variables consisting of joint position and perpendicular offsets of interior control points to the DER rest-state quantities, is differentiable.

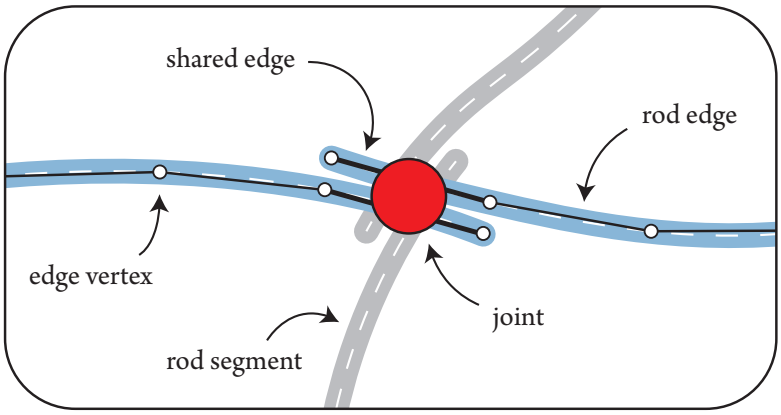


Figure A.2: Terminology of the different elements involved in our C-shell discretization. Consecutive rod segments originate from the same cubic spline, ensuring C^2 smoothness at the joints.

Constant Speed Reparameterization of Interpolating Splines

Arc-length parameterization for cubic spline cannot be expressed analytically, and must be approached numerically. Our reparameterization method consists in fitting an interpolating cubic spline through more interpolated points. We now describe our method and explain why it produces an arc-length parameterized approximation of a given $\mathcal{C}^0$ curve $\gamma : [0,1] \to \mathbb{R}^2$ of length $L(\gamma)$.

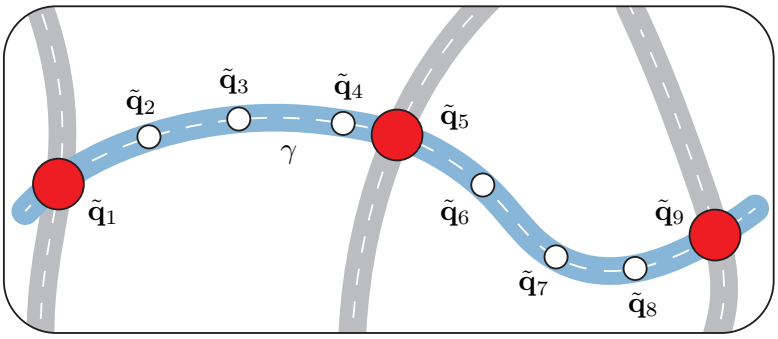


Figure A.3: Notation used.

A curve $\bar{\gamma} : [0,1] \to \mathbb{R}^2$ is constant speed if and only if $\forall s \in [0,1], \quad \|\bar{\gamma}'(s)\| = L(\bar{\gamma})$. We start by sampling $\gamma : [0,1] \to \mathbb{R}^2$ at some locations $(t_i)_{i \in [\![1,n]\!]} \in [0,1]^n$, and call the sampled points *refinement points*: $\tilde{\mathbf{q}}_i := \gamma(t_i)$. We assume that the locations are such that $t_1{=}0$, $t_n{=}1$, and $(t_i)_{i \in [\![1,n]\!]}$ is sorted in an increasing fashion. The length of the polyline joining refinement points up to i is defined as $\tilde{L}_i := \sum_{k=2}^{i} \|\tilde{\mathbf{q}}_k - \tilde{\mathbf{q}}_{k-1}\|$, and $\tilde{s}_i := \tilde{L}_i / \tilde{L}_n$. We fit an interpolating cubic

spline $\tilde{\gamma}_n$ (length $L(\tilde{\gamma}_n)$) to the set of refinement points using the knots $\tilde{s}_i$. Hence, $\tilde{\gamma}_n(\tilde{s}_i) = \tilde{\mathbf{q}}_i$, and we may compute

$$\frac{\tilde{\gamma}_n(\tilde{s}_{i+1}) - \tilde{\gamma}_n(\tilde{s}_i)}{\tilde{s}_{i+1} - \tilde{s}_i} = \frac{\tilde{\mathbf{q}}_{i+1} - \tilde{\mathbf{q}}_i}{\|\tilde{\mathbf{q}}_{i+1} - \tilde{\mathbf{q}}_i\|} \sum_{k=2}^{n} \|\tilde{\mathbf{q}}_k - \tilde{\mathbf{q}}_{k-1}\|,$$

so that the norm of the above quantity equals $\tilde{L}_n$. In the limit of thinner increments $\Delta t := \max_{i\in[\![2,n]\!]}(t_i - t_{i-1})$, and large number of samples n, the length of the polyline $\tilde{L}_n$ converges to the length of the cubic spline $L(\tilde{\gamma}_n)$, which converges to the length of the initial curve $L(\gamma)$. Indeed, the cubic spline $\tilde{\gamma}_n$ approximates γ up to the fourth order on $\Delta\tilde{s} := \max_{i\in[\![2,n]\!]}(\tilde{s}_i - \tilde{s}_{i-1})$ [Hall and Meyer, 1976], and, by continuity of γ, $\Delta\tilde{s}$ converges to 0 as Δt goes to 0.

Therefore, for any $\epsilon > 0$, we may find a fine subdivision $(t_i)_{i\in[\![1,n]\!]}$ for a large enough number of samples n such that for any $\tilde{s}_i \leq \tilde{s} < \tilde{s}_{i+1}$

$$\begin{aligned}
|\|\tilde{\gamma}_n'(s)\| - L| &\leq \left| \|\tilde{\gamma}_n'(s)\| - \frac{\|\tilde{\gamma}_n(\tilde{s}_{i+1}) - \tilde{\gamma}_n(\tilde{s}_i)\|}{\tilde{s}_{i+1} - \tilde{s}_i} \right| \\
&\qquad + \left| \frac{\|\tilde{\gamma}_n(\tilde{s}_{i+1}) - \tilde{\gamma}_n(\tilde{s}_i)\|}{\tilde{s}_{i+1} - \tilde{s}_i} - \tilde{L}_n \right| + |\tilde{L}_n - L| \\
&\leq \epsilon + 0 + \epsilon.
\end{aligned}$$

The first upper bound comes from the $\mathcal{C}^1$ continuity of cubic splines on their domain, while the last upper bound comes from our previous argument about polyline length convergence. This proves that, in the limit of large refinements, our strategy produces an arc-length parameterized cubic spline that approximates γ. Increasing the total number of refinement points n improves the reparameterization at the expense of higher computational complexity.

A.5 Planarization

We describe our planarization algorithm and support our observations on the deployment kinetics on an actual C-shell design. The algorithm takes a B-spline surface then jointly optimizes the joints positions in the flat state and on the surface under feasibility constraints.

Variables

In this part, a C-shell is represented by its joints $\mathbf{c}_i \in \mathbb{R}^2$ and each of its constitutive rod is simplified into a polyline connecting the joints. To each joint in the flat state, we associate a position on the target surface given by the surface parameters $(u_i, v_i) \in [0,1]^2$. The surface is given by the differentiable function $S : [0,1]^2 \to \mathbb{R}^3$. We jointly optimize the joints positions in the flat state

$\mathbf{c}_i$ and the corresponding target joint positions in the deployed state (u_i, v_i) so that it minimizes deviations to a set of kinetic rules we observed in some designs.

Objective

Our total planarization objective $\mathcal{E}(\mathbf{c}_1, \dots, \mathbf{c}_n, u_1, v_1, \dots, u_n, v_n)$ is composed of four terms we describe later

$$\mathcal{E} = \mathcal{E}_{\text{len}} + \mathcal{E}_{\text{eq}} + \mathcal{E}_{\alpha} + \mathcal{E}_{\text{bnd}},$$

where we omit dependencies of the different terms to the variables for clarity.

Segments Lengths Preservation

We first observe that stretching and twisting are often negligible compared to the bending energy in the deployed state for the cross-sections used. We hence assume the polyline edges to be inextensible. The distances between neighboring joints should be preserved in the flat and deployed states. We hence define the segments lengths preservation term as

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

where $\bar{l}_0$ is the surface diameter divided by the number of edges in the linkage, and w_{len} is a weight that controls the importance given to that term. The first sum is taken over each curve c in the linkage, and the second is taken over the joints connected by the polylines. We denote by $\mathbf{c}_{c,i}$ the i-th joint along curve c, similar for $(u_{c,i}, v_{c,i})$.

Deployed State Equilibrium (Straight Linkage)

The deployed state we obtain from the parameters (u_i, v_i) may not be at equilibrium given the rest quantities extracted from the flat joints positions $\mathbf{c}_i$. We simulate our linkage made of polylines as an ensemble of DERs connected at the joints. Each curve of the C-shell is approximated by a single DER whose vertices are given by the interpolated joints, and edges are the rod segments.

The simplified linkage has rest quantities given by $\mathbf{p}(\mathbf{c})$, where $\mathbf{c}$ holds all the flat joints positions, and deformed joints positions given by $\bar{\mathbf{c}}(u_1, v_1, \dots, u_n, v_n)$. For some material frame angles $\boldsymbol{\theta}$, the total energy of the deployed linkage is given by $E(\bar{\mathbf{c}}, \boldsymbol{\theta}, \mathbf{p}) := \sum_c E_c(\bar{\mathbf{c}}_c, \boldsymbol{\theta}_c, \mathbf{p}_c)$, where we sum over all DERs c. The average opening angle in the deployed state can be computed thanks

to the function $\bar{\alpha}(\bar{\mathbf{c}})$. We set the material frame angles such that the structure is as equilibrium when holding the deployed joints in place, and solve

$$\boldsymbol{\theta}^*(\bar{\mathbf{c}}, \mathbf{p}) = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, E(\bar{\mathbf{c}}, \boldsymbol{\theta}, \mathbf{p}).$$

Sensitivities of the material frame angles at equilibrium to the deployed joints positions and rest quantities can be obtained as

$$\frac{\partial^2 E}{\partial \boldsymbol{\theta}^2}\begin{bmatrix}\dfrac{\partial \boldsymbol{\theta}^*}{\partial \bar{\mathbf{c}}} & \dfrac{\partial \boldsymbol{\theta}^*}{\partial \mathbf{p}}\end{bmatrix} = -\begin{bmatrix}\dfrac{\partial^2 E}{\partial \boldsymbol{\theta}\partial \bar{\mathbf{c}}} & \dfrac{\partial^2 E}{\partial \boldsymbol{\theta}\partial \mathbf{p}}\end{bmatrix}.$$

Denoting by λ the torque applied uniformly at the joints of the structure, we measure equilbrium based on the force balance equation

$$\hat{\mathcal{E}}_{\text{eq}}(\bar{\mathbf{c}}, \boldsymbol{\theta}, \lambda, \mathbf{p}) := \frac{w_{\text{eq}}}{2(YA_0)^2}\left\|\frac{\partial E}{\partial \bar{\mathbf{c}}} - \lambda\frac{\partial \bar{\alpha}}{\partial \bar{\mathbf{c}}}\right\|^2,$$

where w_{eq} controls the importance granted to that criterion compared to other terms in the planarization objective, and the Young modulus Y and the cross section area A_0 scale the force values. The optimal torque can be computed analytically as

$$\lambda^*(\bar{\mathbf{c}}, \boldsymbol{\theta}, \mathbf{p}) = \frac{\dfrac{\partial E}{\partial \bar{\mathbf{c}}}\cdot\dfrac{\partial \bar{\alpha}}{\partial \bar{\mathbf{c}}}^{\mathsf{T}}}{\left\|\dfrac{\partial \bar{\alpha}}{\partial \bar{\mathbf{c}}}\right\|^2}.$$

We hence define the equilibrium criterion as

$$\mathcal{E}_{\text{eq}}(\bar{\mathbf{c}}, \mathbf{p}) := \hat{\mathcal{E}}_{\text{eq}}\left(\bar{\mathbf{c}}, \boldsymbol{\theta}^*(\bar{\mathbf{c}}, \mathbf{p}), \lambda^*(\bar{\mathbf{c}}, \boldsymbol{\theta}^*(\bar{\mathbf{c}}, \mathbf{p}), \mathbf{p}), \mathbf{p}\right).$$

The gradients with respect to the deployed positions can be obtained as

$$\begin{aligned}\frac{\partial \mathcal{E}_{\text{eq}}}{\partial \bar{\mathbf{c}}} &= \frac{\partial \hat{\mathcal{E}}_{\text{eq}}}{\partial \bar{\mathbf{c}}} + \frac{\partial \hat{\mathcal{E}}_{\text{eq}}}{\partial \boldsymbol{\theta}}\frac{\partial \boldsymbol{\theta}^*}{\partial \bar{\mathbf{c}}} + \cancelto{0}{\frac{\partial \hat{\mathcal{E}}_{\text{eq}}}{\partial \lambda}}\left(\frac{\partial \lambda^*}{\partial \bar{\mathbf{c}}} + \frac{\partial \lambda^*}{\partial \boldsymbol{\theta}}\frac{\partial \boldsymbol{\theta}^*}{\partial \bar{\mathbf{c}}}\right)\\ &= \frac{w_{\text{eq}}}{2(YA_0)^2}\left(\frac{\partial E}{\partial \bar{\mathbf{c}}} - \lambda^*\frac{\partial \bar{\alpha}}{\partial \bar{\mathbf{c}}}\right)\left(\frac{\partial^2 E}{\partial \bar{\mathbf{c}}^2} - \lambda^*\frac{\partial^2 \bar{\alpha}}{\partial \bar{\mathbf{c}}^2} + \frac{\partial^2 E}{\partial \bar{\mathbf{c}}\partial \boldsymbol{\theta}}\frac{\partial \boldsymbol{\theta}^*}{\partial \bar{\mathbf{c}}}\right)\\ &= \frac{w_{\text{eq}}}{2(YA_0)^2}\left[\left(\frac{\partial E}{\partial \bar{\mathbf{c}}} - \lambda^*\frac{\partial \bar{\alpha}}{\partial \bar{\mathbf{c}}}\right)\left(\frac{\partial^2 E}{\partial \bar{\mathbf{c}}^2} - \lambda^*\frac{\partial^2 \bar{\alpha}}{\partial \bar{\mathbf{c}}^2}\right) + \mathbf{y}_{\text{eq}}^{\mathsf{T}}\frac{\partial^2 E}{\partial \boldsymbol{\theta}\partial \bar{\mathbf{c}}}\right],\end{aligned}$$

where the adjoint state vector $\mathbf{y}_{\text{eq}}$ satisfies the following linear system

$$\frac{\partial^2 E}{\partial \boldsymbol{\theta}^2}\mathbf{y}_{\text{eq}} = -\left(\frac{\partial E}{\partial \bar{\mathbf{c}}} - \lambda^*\frac{\partial \bar{\alpha}}{\partial \bar{\mathbf{c}}}\right)^{\mathsf{T}}.$$

Similarly, the gradient with respect to the rest quantities reads

$$\begin{aligned}\frac{\partial \mathcal{E}_{\text{eq}}}{\partial \mathbf{p}} &= \frac{\partial \hat{\mathcal{E}}_{\text{eq}}}{\partial \mathbf{p}} + \frac{\partial \hat{\mathcal{E}}_{\text{eq}}}{\partial \boldsymbol{\theta}}\frac{\partial \boldsymbol{\theta}^*}{\partial \mathbf{p}} + \cancelto{0}{\frac{\partial \hat{\mathcal{E}}_{\text{eq}}}{\partial \lambda}}\left(\frac{\partial \lambda^*}{\partial \mathbf{p}} + \frac{\partial \lambda^*}{\partial \boldsymbol{\theta}}\frac{\partial \boldsymbol{\theta}^*}{\partial \mathbf{p}}\right) \\ &= \frac{w_{\text{eq}}}{2(YA_0)^2}\left[\left(\frac{\partial E}{\partial \mathbf{p}} - \lambda^*\frac{\partial \bar{\alpha}}{\partial \mathbf{p}}\right)\frac{\partial^2 E}{\partial \mathbf{p}^2} + \mathbf{y}_{\text{eq}}^{\mathsf{T}}\frac{\partial^2 E}{\partial \boldsymbol{\theta}\partial \mathbf{p}}\right].\end{aligned}$$

The gradient with respect to the flat joints positions are then obtained through autodifferentiation using PyTorch. We use the analytical B-spline surface derivatives to backpropagate $\frac{\partial \mathcal{E}}{\partial \bar{\mathbf{c}}}$ to the gradient with respect the surface parameters (u_i, v_i).

Opening Angles Increment Spread

In our designs, we also observed that the opening angles were all either opening or closing as the linkage deploys. We prevent opening angles to open and close at the same time in the linkage by reducing the variance of the opening angle increments. We define the signed opening angle of a quadrilateral q as $\alpha_q := \angle(\mathbf{c}_{q,2}-\mathbf{c}_{q,1}, \mathbf{c}_{q,4}-\mathbf{c}_{q,1})$, where the 4 vertices are numbered consistently accross the quadrilaterals in the linkage. Similar for the opening angles in the deployed states $\bar{\alpha}_q$. We define the opening angle increments as $\Delta\alpha_q = \bar{\alpha}_q - \alpha_q$, and our term reads

$$\mathcal{E}_\alpha(\mathbf{c}_1, \ldots, \mathbf{c}_n, u_1, v_1, \ldots, u_n, v_n) := \frac{w_\alpha}{2}\mathbb{V}\text{ar}_q\left[\Delta\alpha_q\right].$$

Soft Boundary Joint Pinning

The last term softly enforces pinning constraints to boundary joints $(u_{b,i}, v_{b,i})$ to some user-defined positions $\bar{\mathbf{c}}_{b,i}^{(\text{tar})} \in \mathbb{R}^3$ as

$$\mathcal{E}_{\text{bnd}}(u_1, v_1, \ldots, u_n, v_n) := \frac{w_{\text{bnd}}}{2\ \bar{l}_0^{\ 2}}\sum_i \left\| S(u_{b,i}, v_{b,i}) - \bar{\mathbf{c}}_{b,i}^{(\text{tar})} \right\|^2,$$

where the sum is taken over the boundary joints, w_{bnd} enables tuning the importance given to that term, and $\bar{l}_0$ is the previously defined length scale.

Constraints

We first constrain the deployed joints variables (u_i, v_i) to lie within the range $[0, 1]$. Second, the quadrilaterals should not self-intersect in the flat state. For that, we triangulate the quadrilaterals in two ways and compute the signed areas of the triangles, see Figure A.4. We require that at least one of the two triangulations of a quadrilateral q has two positively oriented triangles:

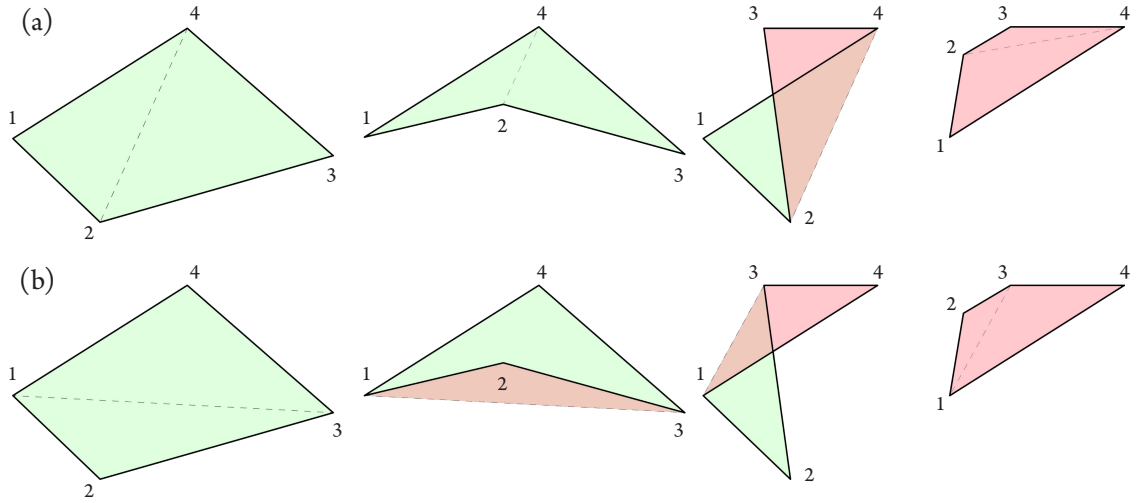


Figure A.4: Self-intersection detection for quadrilaterals. We triangulate the quadrilateral in two ways (rows (a) and (b)) and compute the signed area of the triangles. The dashed lines represent the quadrilateral's triangulation. Checking that at least one triangulation produces two positively oriented triangles ensures the transformed quadrilateral to be non-self-intersecting.

$$\max(\min(\mathrm{SA}_{q,412}, \mathrm{SA}_{q,234}), \min(\mathrm{SA}_{q,123}, \mathrm{SA}_{q,341})) > \epsilon_{\text{area}}|S|, \tag{A.1}$$

where $\mathrm{SA}_{q,ijk}$ computes the signed area of the triangle of vertices i, j, k in quadrilateral q, and $|S|$ is the area of the target surface. ϵ_{area} determines how far we want the quadrilaterals from being self-intersecting.

A.6 Design Optimization

Our design optimization consists in minimizing the function $\bar{J}$ of the curves degrees of freedom and average opening angle $\bar{\mathbf{q}}$

$$\bar{J}(\bar{\mathbf{q}}) := J\left(\mathbf{q}_{\text{ip}}(\mathbf{q}), \bar{\mathbf{p}}(\bar{\mathbf{q}})\right), \quad J(\mathbf{q}_{\text{ip}}, \bar{\mathbf{p}}) := \frac{1}{E_0} E(\bar{\mathbf{x}}^*_{\text{3D}}(\bar{\mathbf{p}}), \mathbf{p}) + T_t(\bar{\mathbf{x}}^*_{\text{3D}}(\bar{\mathbf{p}})) + R(\mathbf{q}_{\text{ip}}, \mathbf{p}), \tag{A.2}$$

where $\mathbf{q}_{\text{ip}}$ are the intermediate interpolated points constructed from the joints positions and orthogonal offsets stored in $\mathbf{q}$.

Fairness Regularization

The last term in the definition of J is a regularization that we split into two parts

$$R(\mathbf{q}_{\text{ip}}, \mathbf{p}) := \underbrace{\frac{w_{\text{ip}}}{2l_0^2} \left\| \mathbf{L}_{\text{ip}} \mathbf{q}_{\text{ip}} - \mathbf{q}_{\text{ip,f}} \right\|_2^2}_{w_{\text{ip}} R_{\text{ip}}(\mathbf{q}_{\text{ip}})} + \underbrace{\frac{w_\kappa}{2\bar{\kappa}_0} \mathbf{p}_\kappa^\mathsf{T} \mathbf{L} \mathbf{p}_\kappa}_{w_\kappa R_\kappa(\mathbf{p})}. \tag{A.3}$$

The first term is a fairness term that measures the Dirichlet energy of the directed graph of interpolated points. We build this graph by first connecting the interpolated points $\mathbf{q}_{\text{ip}}$ and their neighbors along each curve from the curve linkage. Then, we remove the edge going from inner interpolated points to either one of each curve endpoints. This prevents the boundary from shrinking. Interpolated points with no inward pointing edges are labeled as *fixed* and are stored in the constant vector $\mathbf{q}_{\text{ip,f}} \in \mathbb{R}^{2n_{\text{ip}}}$. This vector also contains 0s wherever the associated interpolated point is not fixed. The user may remove edges to fix additional interpolated points. $\mathbf{L}_{\text{ip}}$ is the uniform Laplacian matrix associated to the interpolated points graph.

The second term penalizes the deviation of the rest curvature at a vertex to the average of its two neighboring vertices along each rod. We use the average of the rest lengths of the rod segments $\bar{\kappa}_0$ to scale the regularization term. We define $\mathbf{p}_\kappa$ to extract the rest curvatures from the vector of rest quantities $\mathbf{p}$, and $\mathbf{L}$ as the 1D uniform Laplacian matrix.

The coefficients w_{ip} and w_κ control the importance given to each of the two terms.

Gradients and Hessian-Vector Products

The objective $\bar{J}$ is minimized using an off-the-shelf Sequential Linear-Quadratic Programming (SLQP) algorithm provided by Knitro [Waltz and Nocedal, 2004]. The EQP part is solved using a Newton-CG like approach, which requires computing gradients and Hessian-Vector Products (HVPs) of the objective with respect to the curves degrees of freedom $\bar{\mathbf{q}}$. This vector also includes the target average opening angle $\bar{\alpha}_{\mathbf{q}}$. We first expose the gradients and HVPs with respect to the rest quantities $\mathbf{p}$, and $\bar{\alpha}_{\mathbf{p}}$. Then, we explain how they relate to the same quantities expressed with respect to design parameters.

Rest Quantities

We define an objective function that depends on the interpolated points, the deployed degrees of freedom, and the rest quantities as

$$\tilde{J}(\mathbf{q}_{\text{ip}}, \bar{\mathbf{x}}_{\text{3D}}, \bar{\mathbf{p}}) := \frac{1}{E_0} E(\bar{\mathbf{x}}_{\text{3D}}, \mathbf{p}) + T_t(\mathbf{x}_{\text{3D}}) + R(\mathbf{q}_{\text{ip}}, \mathbf{p}).$$

The previously defined objective as a function of the rest quantities can be written as $J(\mathbf{q}_{\text{ip}}, \bar{\mathbf{p}}) = \tilde{J}(\mathbf{q}_{\text{ip}}, \bar{\mathbf{x}}^*_{\text{3D}}(\bar{\mathbf{p}}), \bar{\mathbf{p}})$. We express the first-order KKT conditions for the deployed equilibrium problem $\bar{\mathbf{x}}^*_{\text{3D}}(\bar{\mathbf{p}}) := \text{argmin}_{\bar{\mathbf{x}}}\, \bar{E}(\bar{\mathbf{x}}, \mathbf{p})$ subject to $\bar{\alpha}_{\mathbf{x}} = \bar{\alpha}_{\mathbf{p}}$ as

$$\begin{cases} \left.\dfrac{\partial \bar{E}}{\partial \bar{\mathbf{x}}}\right|_{\bar{\mathbf{x}}^*_{\text{3D}}(\bar{\mathbf{p}}), \bar{\mathbf{p}}} + \lambda(\bar{\mathbf{p}})\mathbf{e}_{\bar{\alpha}}^{\mathsf{T}} = \mathbf{0} \\ \bar{\alpha}^*_{\mathbf{x}}(\bar{\mathbf{p}}) = \bar{\alpha}_{\mathbf{p}}, \end{cases}$$

where $\lambda(\bar{\mathbf{p}})$ is the Lagrange multiplier of the equality constrained minimization problem. Differentiating gives the following KKT system

$$\begin{cases} \mathbf{H}_{\text{3D}} \dfrac{\partial \mathbf{x}^*_{\text{3D}}}{\partial \bar{\mathbf{p}}} = -\begin{bmatrix} \dfrac{\partial^2 \bar{E}}{\partial \mathbf{x} \partial \mathbf{p}} & \dfrac{\partial^2 \bar{E}}{\partial \mathbf{x} \partial \bar{\alpha}_{\mathbf{x}}} \end{bmatrix} \\ \dfrac{\partial^2 \bar{E}}{\partial \bar{\alpha}_{\mathbf{x}} \partial \mathbf{x}} \dfrac{\partial \mathbf{x}^*_{\text{3D}}}{\partial \bar{\mathbf{p}}} + \begin{bmatrix} \dfrac{\partial^2 \bar{E}}{\partial \bar{\alpha}_{\mathbf{x}} \partial \mathbf{p}} & \dfrac{\partial^2 \bar{E}}{\partial \bar{\alpha}_{\mathbf{x}} \partial \bar{\alpha}_{\mathbf{x}}} \end{bmatrix} = -\dfrac{\partial \lambda}{\partial \bar{\mathbf{p}}} \\ \dfrac{\partial \bar{\alpha}^*_{\mathbf{x}}}{\partial \mathbf{p}} = \mathbf{0} \\ \dfrac{\partial \bar{\alpha}^*_{\mathbf{x}}}{\partial \bar{\alpha}_{\mathbf{p}}} = 1, \end{cases} \tag{A.4}$$

where $\mathbf{H}_{\text{3D}} := \frac{\partial^2 \bar{E}}{\partial \mathbf{x}^2}(\bar{\mathbf{x}}^*_{\text{3D}}(\bar{\mathbf{p}}), \mathbf{p})$ is the Hessian of the elastic energy and surface attraction term of the deployed linkage with respect to the rod linkage degrees of freedom, and $\bar{\mathbf{p}} := [\mathbf{p}^{\mathsf{T}}\ \bar{\alpha}_t]^{\mathsf{T}}$. We then define the adjoint state vector $\mathbf{w}$ as

$$\mathbf{H}_{\text{3D}} \mathbf{w} = \frac{\partial \tilde{J}}{\partial \mathbf{x}}^{\mathsf{T}},$$

where the gradient $\frac{\partial \tilde{J}}{\partial \mathbf{x}}$ can be computed analytically using the elastic rods model, see [Panetta et al., 2019] for details. The overall gradient of the objective J with respect to the rest quantities can be written as

$$\frac{\partial J}{\partial \bar{\mathbf{p}}} = -\mathbf{w}^{\mathsf{T}} \begin{bmatrix} \dfrac{\partial^2 \bar{E}}{\partial \mathbf{x} \partial \mathbf{p}} & \dfrac{\partial^2 \bar{E}}{\partial \mathbf{x} \partial \bar{\alpha}_{\mathbf{x}}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \dfrac{\partial \tilde{J}}{\partial \bar{\alpha}_{\mathbf{x}}} \end{bmatrix} + \frac{\partial \tilde{J}}{\partial \bar{\mathbf{p}}}.$$

The derivative of the objective J with respect to the interpolated points is computed analytically as $\frac{\partial J}{\partial \mathbf{q}_{\text{ip}}} = \beta_{\text{ip}} \frac{\partial R_{\text{ip}}}{\partial \mathbf{q}_{\text{ip}}}$.

We now compute the HVP for J along a given direction $\delta\bar{\mathbf{p}}$. First, we write the KKT system in Equation (A.4) along that direction to obtain

$$\delta \bar{\mathbf{x}}^*_{\text{3D}} = \begin{bmatrix} -\mathbf{H}_{\text{3D}}^{-1} \left(\dfrac{\partial^2 \bar{E}}{\partial \mathbf{x} \partial \mathbf{p}} \delta \mathbf{p} + \dfrac{\partial^2 \bar{E}}{\partial \mathbf{x} \partial \bar{\alpha}_{\mathbf{x}}} \delta \bar{\alpha}_{\mathbf{p}} \right) \\ \delta \bar{\alpha}_{\mathbf{p}} \end{bmatrix}, \tag{A.5}$$

where we define for short $\delta\bar{\mathbf{x}}^*_{3\mathrm{D}} := \frac{\partial \bar{\mathbf{x}}^*_{3\mathrm{D}}}{\partial \bar{\mathbf{p}}}\delta\bar{\mathbf{p}}$. Similarly, the perturbation of the adjoint state vector $\delta\mathbf{w} := \frac{\partial \mathbf{w}}{\partial \bar{\mathbf{p}}}\delta\bar{\mathbf{p}}$ must satisfy the linear system given by

$$\mathbf{H}_{3\mathrm{D}}\delta\mathbf{w} = \frac{\partial^2 \tilde{J}}{\partial \mathbf{x}\partial\bar{\mathbf{x}}}\delta\bar{\mathbf{x}}^*_{3\mathrm{D}} + \frac{\partial^2 \tilde{J}}{\partial \mathbf{x}\partial\mathbf{p}}\delta\mathbf{p} - \left(\frac{\partial^3 \bar{E}}{\partial\mathbf{p}\partial\mathbf{x}\partial\bar{\mathbf{x}}}\delta\bar{\mathbf{x}}^*_{3\mathrm{D}} + \frac{\partial^3 \bar{E}}{\partial\mathbf{p}\partial\mathbf{x}\partial\mathbf{p}}\delta\mathbf{p}\right)\mathbf{w}.$$

The full HVP can then be written as

$$\begin{aligned}\frac{\partial^2 J}{\partial \bar{\mathbf{p}}^2}\delta\bar{\mathbf{p}} = -\begin{bmatrix}\dfrac{\partial^2 \bar{E}}{\partial\mathbf{p}\partial\mathbf{x}} \\ \dfrac{\partial^2 \bar{E}}{\partial\bar{\alpha}_{\mathbf{x}}\partial\mathbf{x}}\end{bmatrix}\delta\mathbf{w} - \begin{bmatrix}\dfrac{\partial^3 \bar{E}}{\partial\mathbf{p}\partial\mathbf{x}\partial\bar{\mathbf{x}}}\delta\bar{\mathbf{x}}^*_{3\mathrm{D}} + \dfrac{\partial^3 \bar{E}}{\partial\mathbf{p}\partial\mathbf{x}\partial\mathbf{p}}\delta\mathbf{p} \\ \dfrac{\partial^3 \bar{E}}{\partial\bar{\alpha}_{\mathbf{x}}\partial\mathbf{x}\partial\bar{\mathbf{x}}}\delta\bar{\mathbf{x}}^*_{3\mathrm{D}} + \dfrac{\partial^3 \bar{E}}{\partial\bar{\alpha}_{\mathbf{x}}\partial\mathbf{x}\partial\mathbf{p}}\delta\mathbf{p}\end{bmatrix}\mathbf{w} \\ + \begin{bmatrix}\mathbf{0} \\ \dfrac{\partial^2 \tilde{J}}{\partial\bar{\alpha}_{\mathbf{x}}\partial\bar{\mathbf{x}}}\delta\bar{\mathbf{x}}^*_{3\mathrm{D}} + \dfrac{\partial^2 \tilde{J}}{\partial\bar{\alpha}_{\mathbf{x}}\partial\bar{\mathbf{p}}}\delta\bar{\mathbf{p}}\end{bmatrix} \\ + \frac{\partial^2 \tilde{J}}{\partial\bar{\mathbf{p}}\partial\bar{\mathbf{x}}}\delta\bar{\mathbf{x}}^*_{3\mathrm{D}} + \frac{\partial^2 \tilde{J}}{\partial\bar{\mathbf{p}}\partial\bar{\mathbf{p}}}\delta\bar{\mathbf{p}}.\end{aligned} \tag{A.6}$$

The Hessian of J with respect to the control points is simply given by $\beta_{\mathrm{ip}}/l_0^2\,\mathbf{L}_{\mathrm{ip}}^{\mathsf{T}}\mathbf{L}_{\mathrm{ip}}$. Third order derivatives of the energy are obtained through automatic differentiation as explained in [Panetta et al., 2019].

Design Parameters

Gradients of $\bar{J}(\bar{\mathbf{q}})$ can be obtained by backpropagating the gradients of $J(\mathbf{q}_{\mathrm{cp},\bar{\mathbf{p}}})$. In practice, this consists in computing the Vector-Jacobian Product (VJP) and it is performed efficiently using reverse-mode automatic differentiation.

Second order derivatives are more expensive to obtain using automatic differentiation since the computation graph must be traversed multiple times. While constructing the full Hessian $\frac{\partial^2 \bar{J}}{\partial \bar{\mathbf{q}}^2}$ is prohibited, HVPs can be computed using the previously introduced HVPs as

$$\begin{aligned}\frac{\partial^2 \bar{J}}{\partial \bar{\mathbf{q}}^2}\delta\bar{\mathbf{q}} = \frac{\partial \bar{\mathbf{p}}}{\partial \bar{\mathbf{q}}}^{\mathsf{T}}\frac{\partial^2 J}{\partial \bar{\mathbf{p}}^2}\frac{\partial \bar{\mathbf{p}}}{\partial \bar{\mathbf{q}}}\delta\bar{\mathbf{q}} + \frac{\partial}{\partial \bar{\mathbf{q}}}\left(\frac{\partial}{\partial \bar{\mathbf{q}}}\left(\mathbf{g}_{\bar{\mathbf{p}}}^{\mathsf{T}}\cdot\bar{\mathbf{p}}\right)\cdot\delta\bar{\mathbf{q}}\right) + \frac{\partial \mathbf{q}_{\mathrm{ip}}}{\partial \bar{\mathbf{q}}}^{\mathsf{T}}\frac{\partial^2 J}{\partial \mathbf{q}_{\mathrm{ip}}{}^2}\frac{\partial \mathbf{q}_{\mathrm{ip}}}{\partial \bar{\mathbf{q}}}\delta\bar{\mathbf{q}} \\ + \frac{\partial}{\partial \bar{\mathbf{q}}}\left(\frac{\partial}{\partial \bar{\mathbf{q}}}\left(\mathbf{g}_{\mathbf{q}_{\mathrm{ip}}}^{\mathsf{T}}\cdot\mathbf{q}_{\mathrm{ip}}\right)\cdot\delta\bar{\mathbf{q}}\right),\end{aligned} \tag{A.7}$$

where we define the **constant** vectors equal to the current gradients of the objective $\mathbf{g}_{\bar{\mathbf{p}}}^{\mathsf{T}} := \partial J/\partial\bar{\mathbf{p}}$ and $\mathbf{g}_{\mathbf{q}_{\mathrm{ip}}}^{\mathsf{T}} := \partial J/\partial\mathbf{q}_{\mathrm{ip}}$.

Extended Rest Quantities Update

When updating the extended rest quantities by an increment $\delta\bar{\mathbf{p}}$, we estimate the change in the deployed state configuration by truncating the following Taylor expansion up to the second order

$$\bar{\mathbf{x}}^*_{3\mathrm{D}}(\bar{\mathbf{p}}+\delta\bar{\mathbf{p}}) = \bar{\mathbf{x}}^*_{3\mathrm{D}}(\bar{\mathbf{p}}) + \frac{\partial \bar{\mathbf{x}}^*_{3\mathrm{D}}}{\partial \bar{\mathbf{p}}}\delta\bar{\mathbf{p}} + \delta\bar{\mathbf{p}}^{\mathsf{T}}\frac{\partial^2 \bar{\mathbf{x}}^*_{3\mathrm{D}}}{\partial \bar{\mathbf{p}}^2}\delta\bar{\mathbf{p}} + \mathcal{O}(\|\delta\bar{\mathbf{p}}\|^3).$$

The first-order modification of the degrees of freedom can be obtained by solving the system shown in Equation (A.5). The second-order perturbation of the deployed degrees of freedom at equilibrium can be obtained by differentiating Equation (A.5) as shown below:

$$\begin{cases} \mathbf{H}_{3\mathrm{D}}\delta\bar{\mathbf{p}}\dfrac{\partial^2 \mathbf{x}^*_{3\mathrm{D}}}{\partial \bar{\mathbf{p}}^2}\delta\bar{\mathbf{p}} &= -\left[\dfrac{\partial^3 \bar{E}}{\partial \mathbf{x}\partial\bar{\mathbf{x}}\partial\bar{\mathbf{x}}}\delta\bar{\mathbf{x}}^*_{3\mathrm{D}} + \dfrac{\partial^3 \bar{E}}{\partial \mathbf{x}\partial\bar{\mathbf{x}}\partial\mathbf{p}}\delta\mathbf{p}\right]\delta\bar{\mathbf{x}}^*_{3\mathrm{D}} \\ & \qquad -\left[\dfrac{\partial^3 \bar{E}}{\partial \mathbf{x}\partial\mathbf{p}\partial\bar{\mathbf{x}}}\delta\bar{\mathbf{x}}^*_{3\mathrm{D}} + \dfrac{\partial^3 \bar{E}}{\partial \mathbf{x}\partial\mathbf{p}\partial\mathbf{p}}\delta\mathbf{p}\right]\delta\mathbf{p} \\ \delta\bar{\mathbf{p}}\dfrac{\partial^2 \bar{\alpha}^*_{\mathbf{x}}}{\partial \bar{\mathbf{p}}^2}\delta\bar{\mathbf{p}} &= 0. \end{cases}$$

Appendix B

Supplementary Material: Computational Design of a Kit of Parts for Bending-Active Structures

This document provides additional technical details regarding our process for calculating gradients for the preservation energy term $\mathcal{F}$, optimization objective non-dimensionalization, and bound constraints guarantees on the parts. We also show the sets of designs we used for the experiments in the main paper.

B.1 Preservation Energy Optimization

Our preservation energy $\mathcal{F}(\mathbf{x}, \mathbf{r})$ is defined for a physical system with rest variables $\mathbf{r}$ and simulation variables $\mathbf{x}$ as the sum of a target fitting term $\mathcal{T}(\mathbf{x})$ and an elastic energy term $\mathcal{E}(\mathbf{x}, \mathbf{r})$. We aim to efficiently compute the gradient of the preservation energy with respect to the design variables $\mathbf{d}$ (composed of rest variables and Dirichlet constraint values) when evaluated at the equilibrium state $\mathbf{x}^*(\mathbf{d})$ i.e., the gradient of $\tilde{\mathcal{F}}(\mathbf{d}) := \mathcal{F}(\mathbf{x}^*(\mathbf{d}), \mathbf{r})$.

Equilibrium Sensitivities

This term is only evaluated at the equilibrium state $\mathbf{x}^*(\mathbf{d})$ defined as We split the design variables into rest quantities e.g., beam rest lengths, and fixed deformed variables e.g., corner angles, as $\mathbf{d} = [\mathbf{r}_v^\top, \mathbf{r}_f^\top]^\top$.

$$\begin{aligned} \mathbf{x}^*(\mathbf{d}) := \underset{\mathbf{x}}{\operatorname{argmin}}\ & \bar{\mathcal{E}}(\mathbf{x}, \mathbf{r}_v) \\ \text{s.t.} \quad & \mathbf{x}_f = \mathbf{r}_f, \end{aligned} \tag{B.1}$$

where $\bar{\mathcal{E}}(\mathbf{x}, \mathbf{r}_v) = \mathcal{E}(\mathbf{x}, \mathbf{r}_v) + \mathcal{D}(\mathbf{x})$ is the total energy of the system, $\mathcal{D}(\mathbf{x})$ models the external deployment forces, and $\mathbf{x}_f$ are the simulation variables fixed by the Dirichlet constraints $\mathbf{r}_f$. We express the first-order KKT conditions for the equilibrium state as

$$\begin{cases} \left.\dfrac{\partial \bar{\mathcal{E}}}{\partial \mathbf{x}}\right|_{\mathbf{x}^*(\mathbf{d}),\mathbf{d}} + \lambda(\mathbf{d})^\top \dfrac{\partial \mathbf{x}_f}{\partial \mathbf{x}} = \mathbf{0} \\ \mathbf{x}_f^*(\mathbf{d}) = \mathbf{r}_f \end{cases}$$

where λ is the Lagrange multiplier associated with the Dirichlet constraints. We further split the simulation variables into free and fixed components as $\mathbf{x} = [\mathbf{x}_v^\top, \mathbf{x}_f^\top]^\top$. We may remark that

$\frac{\partial \mathbf{x}_f}{\partial \mathbf{x}}$ is a block-diagonal matrix with blocks of zeros and identity matrices, and we differentiate the equilibrium state with respect to the design variables $\mathbf{d}$ as

$$\begin{cases} \mathbf{H}_v \dfrac{\partial \mathbf{x}_v^*}{\partial \mathbf{d}} = -\begin{bmatrix} \dfrac{\partial^2 \bar{\mathcal{E}}}{\partial \mathbf{x}_v \partial \mathbf{r}_v} & \dfrac{\partial^2 \bar{\mathcal{E}}}{\partial \mathbf{x}_v \partial \mathbf{x}_f} \end{bmatrix} \\ \dfrac{\partial^2 \bar{\mathcal{E}}}{\partial \mathbf{x}_f \partial \mathbf{x}_v} \dfrac{\partial \mathbf{x}_v^*}{\partial \mathbf{d}} = -\begin{bmatrix} \dfrac{\partial^2 \bar{\mathcal{E}}}{\partial \mathbf{x}_f \partial \mathbf{r}_v} & \dfrac{\partial^2 \bar{\mathcal{E}}}{\partial {\mathbf{x}_f}^2} \end{bmatrix} - \dfrac{\partial \lambda}{\partial \mathbf{d}} \\ \dfrac{\partial \mathbf{x}_f^*}{\partial \mathbf{d}} = \begin{bmatrix} \dfrac{\partial \mathbf{x}_f^*}{\partial \mathbf{r}_v} & \dfrac{\partial \mathbf{x}_f^*}{\partial \mathbf{r}_f} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix}, \end{cases}$$

where $\mathbf{I}$ is the identity matrix, and $\mathbf{H}_v$ is the Hessian of the total energy with respect to the free simulation variables $\mathbf{x}_v$ evaluated at the equilibrium state. The sensitivity of the equilibrium state $\delta \mathbf{x}^*$ to some design variable perturbation $\delta \mathbf{d}$ can be computed as

$$\delta \mathbf{x}^* = \frac{\partial \mathbf{x}^*}{\partial \mathbf{d}} \delta \mathbf{d} = \begin{bmatrix} -\mathbf{H}_v^{-1} \left(\dfrac{\partial^2 \bar{\mathcal{E}}}{\partial \mathbf{x}_v \partial \mathbf{r}_v} \delta \mathbf{r}_v + \dfrac{\partial^2 \bar{\mathcal{E}}}{\partial \mathbf{x}_v \partial \mathbf{x}_f} \delta \mathbf{r}_f \right) \\ \delta \mathbf{r}_f \end{bmatrix}.$$

Computing the second order perturbation of the equilibrium state given a perturbation of the design variables $\delta \mathbf{d}$ is done by solving the following linear system

$$\begin{cases} \mathbf{H}_v \delta \mathbf{d} \dfrac{\partial^2 \mathbf{x}_v^*}{\partial \mathbf{d}^2} \delta \mathbf{d} &= -\left(\dfrac{\partial^3 \bar{\mathcal{E}}}{\partial \mathbf{x}_v \partial \mathbf{r}_v \partial \mathbf{r}_v} \delta \mathbf{r}_v + \dfrac{\partial^3 \bar{\mathcal{E}}}{\partial \mathbf{x}_v \partial \mathbf{r}_v \partial \mathbf{r}_f} \delta \mathbf{r}_f \right) \delta \mathbf{r}_v \\ & \quad - \left(\dfrac{\partial^3 \bar{\mathcal{E}}}{\partial \mathbf{x}_v \partial \mathbf{r}_f \partial \mathbf{r}_v} \delta \mathbf{r}_v + \dfrac{\partial^3 \bar{\mathcal{E}}}{\partial \mathbf{x}_v \partial \mathbf{r}_f \partial \mathbf{r}_f} \delta \mathbf{r}_f \right) \delta \mathbf{r}_f \\ \delta \mathbf{d} \dfrac{\partial^2 \mathbf{x}_f^*}{\partial \mathbf{d}^2} \delta \mathbf{d} &= 0. \end{cases}$$

Gradients

Using the chain rule, we can compute the gradient of the preservation energy with respect to the design variables as

$$\frac{\partial \tilde{\mathcal{F}}}{\partial \mathbf{d}} = \frac{\partial \mathcal{F}}{\partial \mathbf{x}_v} \frac{\partial \mathbf{x}_v^*}{\partial \mathbf{d}} + \frac{\partial \mathcal{F}}{\partial \mathbf{x}_f} \frac{\partial \mathbf{x}_f^*}{\partial \mathbf{d}} + \frac{\partial \mathcal{F}}{\partial \mathbf{d}} = -\frac{\partial \mathcal{F}}{\partial \mathbf{x}_v} \mathbf{H}_v^{-1} \begin{bmatrix} \dfrac{\partial^2 \bar{\mathcal{E}}}{\partial \mathbf{x}_v \partial \mathbf{r}_v} & \dfrac{\partial^2 \bar{\mathcal{E}}}{\partial \mathbf{x}_v \partial \mathbf{x}_f} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \dfrac{\partial \mathcal{F}}{\partial \mathbf{x}_f} \end{bmatrix} + \frac{\partial \mathcal{F}}{\partial \mathbf{d}},$$

We consequently define the adjoint state vector $\mathbf{w}$ as

$$\mathbf{H}_v \mathbf{w} = \frac{\partial \mathcal{F}}{\partial \mathbf{x}_v}^\top,$$

where the gradient of the preservation energy can be computed analytically from the own physical system's inverse design optimization pipeline. The gradient of $\tilde{\mathcal{F}}$ can thus be computed efficiently using the adjoint state vector $\mathbf{w}$ as

$$\frac{\partial \tilde{\mathcal{F}}}{\partial \mathbf{d}} = -\mathbf{w}^\top \begin{bmatrix} \dfrac{\partial^2 \bar{\mathcal{E}}}{\partial \mathbf{x}_v \partial \mathbf{r}_v} & \dfrac{\partial^2 \bar{\mathcal{E}}}{\partial \mathbf{x}_v \partial \mathbf{x}_f} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \dfrac{\partial \mathcal{F}}{\partial \mathbf{x}_f} \end{bmatrix} + \frac{\partial \mathcal{F}}{\partial \mathbf{d}}.$$

Hessian Vector Products

The Hessian Vector Product (HVP) of the preservation energy given a perturbation of the design variables $\delta\mathbf{d}$ can be computed as

$$\begin{aligned} \frac{\partial^2 \tilde{\mathcal{F}}}{\partial \mathbf{d}^2}\delta\mathbf{d} = &- \begin{bmatrix} \dfrac{\partial^2 \bar{\mathcal{E}}}{\partial \mathbf{r}_v \partial \mathbf{x}_v} \\ \dfrac{\partial^2 \bar{\mathcal{E}}}{\partial \mathbf{x}_f \partial \mathbf{x}_v} \end{bmatrix} \delta\mathbf{w} - \begin{bmatrix} \dfrac{\partial^3 \bar{\mathcal{E}}}{\partial \mathbf{r}_v \partial \mathbf{x}_v \partial \mathbf{x}}\delta\mathbf{x}^* + \dfrac{\partial^3 \bar{\mathcal{E}}}{\partial \mathbf{r}_v \partial \mathbf{x}_v \partial \mathbf{r}_v}\delta\mathbf{r}_v \\ \dfrac{\partial^3 \bar{\mathcal{E}}}{\partial \mathbf{x}_f \partial \mathbf{x}_v \partial \mathbf{x}}\delta\mathbf{x}^* + \dfrac{\partial^3 \bar{\mathcal{E}}}{\partial \mathbf{x}_f \partial \mathbf{x}_v \partial \mathbf{r}_v}\delta\mathbf{r}_v \end{bmatrix} \mathbf{w} \\ &+ \begin{bmatrix} \mathbf{0} \\ \dfrac{\partial^2 \mathcal{F}}{\partial \mathbf{x}_f \partial \mathbf{x}}\delta\mathbf{x}^* + \dfrac{\partial^2 \mathcal{F}}{\partial \mathbf{x}_f \partial \mathbf{r}_v}\delta\mathbf{r}_v \end{bmatrix} + \frac{\partial^2 \mathcal{F}}{\partial \mathbf{d} \partial \mathbf{x}}\delta\mathbf{x}^* + \frac{\partial^2 \mathcal{F}}{\partial \mathbf{d}^2}\delta\mathbf{d}, \end{aligned}$$

where $\delta\mathbf{w}$ is the adjoint state vector perturbation, and satisfies the linear system

$$\mathbf{H}_v \delta\mathbf{w} = \frac{\partial^2 \mathcal{F}}{\partial \mathbf{x}_v \partial \mathbf{x}}\delta\mathbf{x}^* + \frac{\partial^2 \mathcal{F}}{\partial \mathbf{x}_v \partial \mathbf{d}}\delta\mathbf{d} - \left(\frac{\partial^3 \bar{\mathcal{E}}}{\partial \mathbf{x}_v \partial \mathbf{x}_v \partial \mathbf{x}}\delta\mathbf{x}^* + \frac{\partial^3 \bar{\mathcal{E}}}{\partial \mathbf{x}_v \partial \mathbf{x}_v \partial \mathbf{r}_v}\delta\mathbf{r}_v \right) \mathbf{w}.$$

B.2 Non-Dimensionalization of the Preservation Energy

Our preservation energy $\mathcal{F}$ is the sum of the elastic energy of a deformed system $\mathcal{E}$ and a target fitting term $\mathcal{T}$. We non-dimensionalize the elastic energy by scaling it with the Young's modulus Y of the fabrication material times the rest volume V_0 of the system. The bending, twisting, and stretching energies stored in the rods of a system can be derived by plugging strain tensor fields $\boldsymbol{\epsilon}$ induced these deformation modes into the linear elasticity energy $\frac{1}{2}\int_\Omega \boldsymbol{\epsilon} : \mathbf{C} : \boldsymbol{\epsilon}$, where C is the fabrication material's elasticity tensor. For an isotropic material, $\mathbf{C} = Y\mathbf{C_0}(\nu)$, where $\mathbf{C_0}(\nu)$ depends only on the Poisson's ratio ν; for example $\mathbf{C_0}(0)$ is the fourth-order identity tensor. Since strain $\boldsymbol{\epsilon}$ is non-dimensional, this elastic energy clearly is proportional to YV_0. The target fitting term is non-dimensionalized by dividing it by the square of the bounding box diagonal of the target surface.

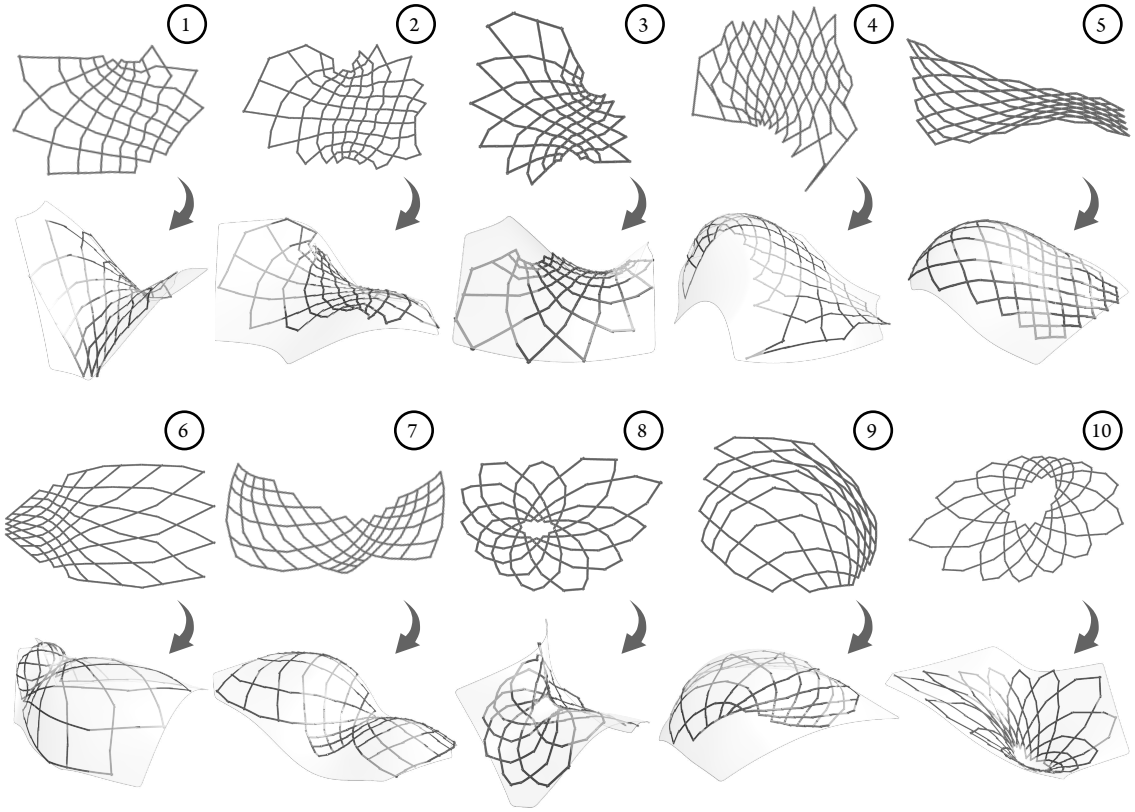


Figure B.1: Conformals Set.

B.3 Bound Constraints Guarantees

Parts within a kit of parts may have to satisfy certain shared fabrication constraints e.g., minimum distance between corners and joints for linear elements of rationalized Orthogonal Grids and C-shells. We assume that such constraints can be expressed as bound constraints on the parts parameters, so that a part i must satisfy $\mathbf{p}_{\min} \leq \mathbf{p}_i \leq \mathbf{p}_{\max}$. We aim to show that any part $\mathbf{p}_i$ obtained from the optimal elements $\mathbf{q}^*$ using the update rule

$$\mathbf{p}_i(\mathbf{q}^*) := \underset{\mathbf{y}}{\operatorname{argmin}} \sum_{j \in c^{-1}(\{i\})} \rho(\mathbf{y}, \mathbf{q}_j^*), \tag{B.2}$$

is guaranteed to satisfy the original feasibility constraints for any assignment $\mathbf{c}$, provided that elements are bound during the optimization process, for the ρ we use in our work.

We impose that elements are bound by the same constraints as the parts, so that $\mathbf{p}_{\min} \leq \mathbf{q}_j \leq \mathbf{p}_{\max}$ for any element j. Under the appropriate symmetry transformations (mirroring), our projection energy ρ is a squared L^2 distance. The solution to the update rule $\mathbf{p}_i(\mathbf{q}^*)$ is given by

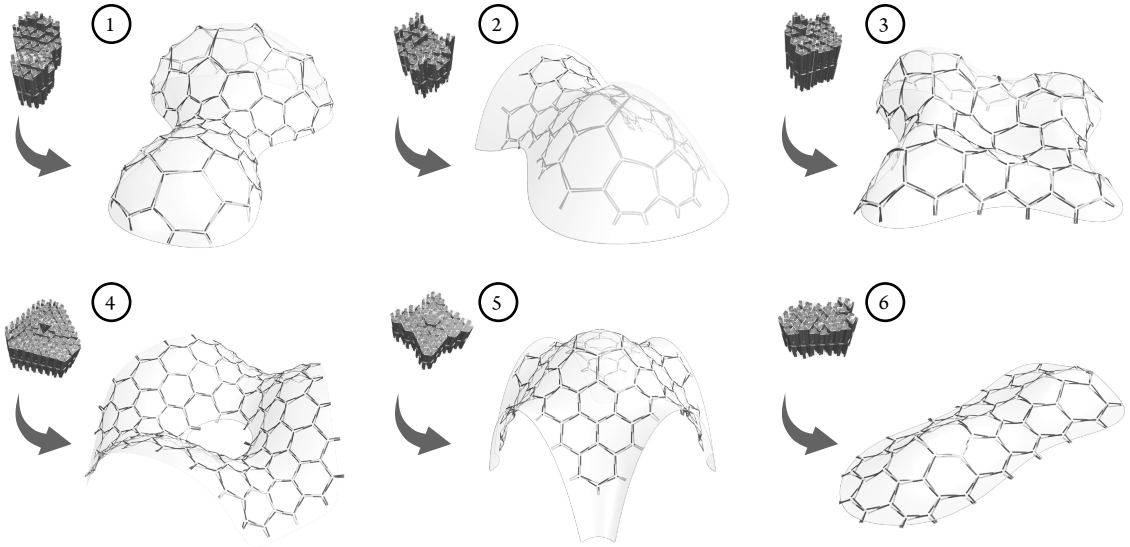


Figure B.2: Umbrella Meshes Set.

the average of the elements assigned to the part. Therefore, the part belongs to the convex hull formed by the elements assigned to it, which itself is included in the axis-aligned bounding box formed by the corners $\mathbf{p}_{\min}$ and $\mathbf{p}_{\max}$. The part $\mathbf{p}_i(\mathbf{q}^*)$ is thus guaranteed to satisfy the original feasibility constraints. Note that unassigned parts can trivially be set to the mean of the corners $\mathbf{p}_{\min}$ and $\mathbf{p}_{\max}$ by default.

B.4 Datasets

We now show the sets of designs we used for the experiments in the main paper.

Conformals

The 10 designs of the Conformals dataset shown in Figure B.1 are obtained following the methodology presented in [Becker et al., 2024]. The designs are conformal deformations of a base grid with an annulus (8 and 10) or a regular (the others) topology. Layout variations are obtained by applying a conformal map defined by a boundary a user can manipulate as presented in [Eck et al., 1995] and [Sawhney and Crane, 2017]. The target surface is extracted by smoothly interpolating the joints of the deployed C-shell.

We reuse the same surfaces for the orthogonal grids. They are initialized by either tracing uv-isolines on the target surface and converting them into piecewise straight beams or by connecting the projections of the joints on the surface using piecewise straight beams. The designs are further

optimized to better fit the target surface using an inverse design optimization algorithm similar to the one presented in [Becker et al., 2023].

Umbrella Meshes

The 6 designs of the Umbrella Meshes dataset shown in Figure B.2 are obtained from known shapes that have been approximated thanks to the inverse design optimization pipeline presented in [Ren et al., 2022]. The structure topologies i.e., the number of cells and their connectivity, are allowed to vary across experiments.

Appendix C

Abstract of Other Published Work

Here we include the abstract of a publication that was not explicitly discussed in this thesis, but was produced during the same period.

- Mike Yan Michelis, **Quentin Becker**. On Linear Interpolation in the Latent Space of Deep Generative Models. *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*, Spotlight.

  Abstract: The underlying geometrical structure of the latent space in deep generative models is in most cases not Euclidean, which may lead to biases when comparing interpolation capabilities of two models. Smoothness and plausibility of linear interpolations in latent space are associated with the quality of the underlying generative model. In this paper, we show that not all such interpolations are comparable as they can deviate arbitrarily from the shortest interpolation curve given by the geodesic. This deviation is revealed by computing curve lengths with the pull-back metric of the generative model, finding shorter curves than the straight line between endpoints, and measuring a non-zero relative length improvement on this straight line. This leads to a strategy to compare linear interpolations across two generative models. We also show the effect and importance of choosing an appropriate output space for computing shorter curves. For this computation we derive an extension of the pull-back metric.

Photo Credits

Figure 1.1 - courtesy of

- ”British Museum” by y.becart is licensed under CC BY 2.0 (British Museum)
- ”Walt Disney Concert Hall” by Jon Sullivan has been released under CC0 (Walt Disney Concert Hall)
- ”Yas Hotel Abu Dhabi” by kulala13 is licensed under Pixabay Content License (Yas Island Hotel)
- ”IRENA@EXPO2017” by IRENA is licensed under CC BY 2.0 (Nur Alem Museum of Future Energy)
- ”Nordkettenbahn” by W.J.S. is licensed under CC BY 2.0 (Hungerburg funicular)

Figure 1.2 - courtesy of

- ”Frei Otto - Roof for the Mannheim Multihalle 02” by Forgemind ArchiMedia is licensed under CC BY 2.0 (Multihalle)
- ”Weald and Downland Museum at Singleton” by Tim Regan is licensed under CC BY 2.0 (Weald and Downland Gridshell)
- ”The Savill Gardens - Windsor Great Park” by Mark Wordy is licensed under CC BY 2.0 (Savill Building)