Inverse Shape Design with Parametric Representations: Kirchhoff Rods and Parametric Surface Models

by

Christian Hafner

May, 2023

A thesis submitted to the
Graduate School
of the
Institute of Science and Technology Austria
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Committee in charge:

Edouard Hannezo, Chair
Bernd Bickel
Chris Wojtan
Doug L. James
David I.W. Levin

The thesis of Christian Hafner, titled *Inverse Shape Design with Parametric Representations: Kirchhoff Rods and Parametric Surface Models*, is approved by:

Supervisor: Bernd Bickel, ISTA, Klosterneuburg, Austria

Signature: _____

Committee Member: Chris Wojtan, ISTA, Klosterneuburg, Austria

Signature: _____

Committee Member: Doug L. James, Stanford University, Stanford, United States

Signature: _____

Committee Member: David I.W. Levin, University of Toronto, Toronto, Canada

Signature: _____

Defense Chair: Edouard Hannezo, ISTA, Klosterneuburg, Austria

Signature: _____

Signed page is on file

ISTA Thesis, ISSN: 2663-337X

ISBN: 978-3-99078-031-2

I hereby declare that this thesis is my own work and that it does not contain other people's work without this being so stated; this thesis does not contain my previous work without this being stated, and the bibliography contains all the literature that I used in writing the dissertation.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee, and that this thesis has not been submitted for a higher degree to any other university or institution.

I certify that any republication of materials presented in this thesis has been approved by the relevant publishers and co-authors.

Signature: _____

Christian Hafner
May, 2023

Signed page is on file

Abstract

Inverse design problems in fabrication-aware shape optimization are typically solved on discrete representations such as polygonal meshes. This thesis argues that there are benefits to treating these problems in the same domain as human designers, namely, the parametric one. One reason is that discretizing a parametric model usually removes the capability of making further manual changes to the design, because the human intent is captured by the shape parameters. Beyond this, knowledge about a design problem can sometimes reveal a structure that is present in a smooth representation, but is fundamentally altered by discretizing. In this case, working in the parametric domain may even simplify the optimization task. We present two lines of research that explore both of these aspects of fabrication-aware shape optimization on parametric representations.

The first project studies the design of plane elastic curves and Kirchhoff rods, which are common mathematical models for describing the deformation of thin elastic rods such as beams, ribbons, cables, and hair. Our main contribution is a characterization of all curved shapes that can be attained by bending and twisting elastic rods having a stiffness that is allowed to vary across the length. Elements like these can be manufactured using digital fabrication devices such as 3d printers and digital cutters, and have applications in free-form architecture and soft robotics.

We show that the family of curved shapes that can be produced this way admits geometric description that is concise and computationally convenient. In the case of plane curves, the geometric description is intuitive enough to allow a designer to determine whether a curved shape is physically achievable by visual inspection alone. We also present shape optimization algorithms that convert a user-defined curve in the plane or in three dimensions into the geometry of an elastic rod that will naturally deform to follow this curve when its endpoints are attached to a support structure. Implemented in an interactive software design tool, the rod geometry is generated in real time as the user edits a curve and enables fast prototyping.

The second project tackles the problem of general-purpose shape optimization on CAD models using a novel variant of the extended finite element method (XFEM). Our goal is the decoupling between the simulation mesh and the CAD model, so no geometry-dependent meshing or remeshing needs to be performed when the CAD parameters change during optimization. This is achieved by discretizing the embedding space of the CAD model, and using a new high-accuracy numerical integration method to enable XFEM on free-form elements bounded by the parametric surface patches of the model. Our simulation is differentiable from the CAD parameters to the simulation output, which enables us to use off-the-shelf gradient-based optimization procedures. The result is a method that fits seamlessly into the CAD workflow because it works on the same representation as the designer, enabling the alternation of manual editing and fabrication-aware optimization at will.

Acknowledgements

Bernd threw me a lifeline in a precarious situation, for which I want to express my deepest gratitude. He has since given me the liberty to pursue the scientific questions that interest me the most, supported me in pushing for scientific rigor, and has always kept me grounded when research did not go as expected. His advice—technical, career, life—has been invaluable to me, and I could not have wished for a better supervisor, teacher, and colleague.

I would also like to thank Chris, Doug, and David for volunteering their time in shepherding me, and for providing their advice and perspectives on my work. My thanks go also to Moritz for giving me the opportunity to work with him in Zurich, for pushing our project all the way past the finish line, and most vitally, for his appreciation of Wiener Schnitzel and Linzer Torte. Thank you also to Espen and Christian for supporting the project, and Christian for putting together many great (and a great many) movie nights. I'd also like to say thanks to Thomas and Michal for providing emergency paper services to bring it home, and to Stefan and Camille for letting me play a small part in their work. Thank you to Todor and Astrit for their advice and support in all things fabrication. I would also like to express my gratitude to Helmut for the trust he put in me when I needed it, and for his brilliant and inspiring comments.

At ISTA, the list of people who I have grown fond of is long. I would like to thank Ruslan, Ran, Xudong, Martin, "Snack Boy" Desmond, and Yi-Lu for filling our office with life and puns; Manas, Malina, "There and Back Again" Eder, Raphaël, Michal, Thomas, Tomáš, "Yarnmachillo" Georg, Peter, and Sadashige for always lending an ear, for all the fun things we have done in the past few years, and for the gemütliche atmosphere; Kasumi and Camila for being super fun, super silly, and filling my life with music; Elizabeth, Teresa, and "Spearheads" Roshan and Louise for many merry evenings.

Thank you to Marta, Jeff, and Wolfgang for being my family of choice in Vienna. Thank you to Mickaël for his fearless kindness. Thank you to my family for all the ways in which they have loved and encouraged me.

Financial support was provided by the European Research Council (ERC) under grant agreement No. 715767 *"MATERIALIZABLE: Intelligent fabrication-oriented Computational Design and Modeling"*, which I gratefully acknowledge.

About the Author

Christian Hafner was born in 1990 in Linz, Austria and attended the BRG Fadingerstraße High School with a focus on natural sciences and computer science. He moved to Vienna to earn a Bachelor's degree in Software Engineering and a Master's degree in Computer Graphics from TU Wien, where he became interested in shape optimization and digital fabrication. He stayed at TU Wien for two more years as a research assistant, during which he published his first paper on a shape optimization algorithm for tuning natural frequencies of musical instruments. Then, he moved to ISTA to pursue a PhD degree in the group of Bernd Bickel. During his first year, he completed a four-month internship at Disney Research Zürich in the digital fabrication group headed by Moritz Bächer, which lead to the publication of a paper about shape optimization of CAD models, the first work presented in this thesis. In the remaining years of his PhD, he worked on inverse design problems about thin elastic rods, which form the other part of the thesis. In his free time, he plays the piano and organizes a chamber music group at ISTA.

List of Collaborators and Publications

Christian Hafner, Christian Schumacher, Espen Knoop, Thomas Auzinger, Bernd Bickel, and Moritz Bächer. X-cad: Optimizing cad models with extended finite elements. *ACM Trans. Graph.*, 38(6), nov 2019

Christian Hafner and Bernd Bickel. The design space of plane elastic curves. *ACM Trans. Graph.*, 40(4), jul 2021

Christian Hafner and Bernd Bickel. The design space of kirchhoff rods. *Accepted to ACM Trans. Graph. with minor revision*, nov 2022

Table of Contents

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

A recent survey about fabrication-aware design [BFR17] lists seven ways of representing the boundaries of manifold geometry. Polygonal meshes, spectral representations, and level sets of a scalar field on the embedding space are found to be the most commonly used for inverse shape design, in applications ranging from the control of elastic behavior and structural stability to shape approximation and support design. Algebraically defined boundaries, height fields, and subdivision surfaces constitute decidedly less popular representations but are found in applications such as toolpath planning and stably interlocking puzzle pieces. But no representation is as unloved as the *parametric* one, about which the survey remarks disdainfully,

> *"Parametric surfaces [...] lack usability for fabrication applications. For this reason, parametric surfaces are seen in this survey only as input in architectural application, and are almost immediately tessellated."*

In defense of parametric representations for fabrication-aware design, I have to mention at least a work which, in the words of its authors, *"employs parametric surfaces as surface geometry representations at any stage of a computational process to compute self-supporting surfaces"* [MIB15], and a recent paper on the computational design of Weingarten surfaces, which *"found that fourth-order B-splines are well suited to formulate this optimization, since they simplify the evaluation of higher-order derivatives required"* [PKPP21]. I also want to point out two general reasons for which it may be desirable to maintain a parametric representation when faced with fabrication-aware optimization tasks.

The first reason is that there are shape optimization problems to which none of the other representations listed above are suited very well. This is the case for problems in which higher-than-second derivatives of a surface parametrization are relevant [PKPP21], or in which differential constraints can be satisfied exactly using a specific parametric model, whereas other representations will introduce unacceptable discretization errors [TBWP16].

We encounter a problem of the latter kind in Chapters 4 and 5, which are concerned with exploring and understanding the design space of thin elastic rods. There, we discover that collinearity of the inflection points of a plane curve plays a crucial role in interpreting this curve as the equilibrium state of an elastic rod. A parametric curve, such as a spline curve, provides sufficient control over the exact location of inflection points through its control points, while it would be difficult to define—or optimize for—collinearity of inflections on a discrete curve.

The second reason for using parametric representations is that shape optimization is not the last step in a digital fabrication pipeline—fabrication is. And between them, there is usually a number of manual editing steps to integrate the isolated optimized shape with its environment, such as adding interfaces to support structures, partitioning for additive manufacturing, or ornamentation. These tasks are handled in a computer-aided design (CAD) environment, and are significantly simplified if the shape representation is not changed for the sake of the shape optimization phase. We conclude that developing fabrication-aware shape optimization tools that keeps parametric models parametric is a useful goal in itself, even for problems in which other shape representations would simplify the task.

This thesis makes a foray into general-purpose optimization of parametric geometry in Chapter 6, where we develop a framework for discretizing partial differential equations (PDEs) on CAD models in such a way that the solution depends differentiably on the CAD parameters. We use the method to perform PDE-constrained shape optimization directly on these parameters, which gives us the possibility to keep editing the CAD model after optimization in much the same way as before, and thus fits seamlessly into a design-to-fabrication workflow.

The boundary between parametric and other representations is in some sense artificial, as all shape representations depend on parameters. However, the term is often used to distinguish representations in which a smaller number of parameters have an effect that is non-local or difficult to predict by human intuition (such as control points and interpolation weights, or coefficients of a PDE), or cause a qualitative change (such as the degree of a spline) from those in which a large number of uniform parameters have a very direct effect (such as vertex positions, or function values discretized on a grid).

To some degree, the choice of representation reflects how much is known about the solution structure of a problem. If we know next to nothing, we are forced to pick a generic representation with many parameters, because there is no basis for deciding on a specialized, more economic parametrization. Unfortunately, many problems in scientific computing are of this type, such as fluid dynamics, non-linear elasticity, topology optimization, and rendering, at least as far as we know today. In these fields, the only way to approximate a solution quantitatively for arbitrary input data is to solve a heavy computational problem. But even there, mathematical knowledge sometimes translates into performance gains by providing insight into special subproblems, or qualitative information such as regularity of solutions.

Recent examples are topology optimization algorithms that use knowledge about the general solution structure [FMT95] to justify a de-homogenization approach [SOG+22]. For general problems in elasticity, regularity knowledge can inform the choice of simulation method, for example to describe the displacement field around a stress singularity with fewer parameters, and knowledge of analytical solutions can be used in animation to avoid heavy-duty simulations [DGJ18]. Sometimes, solutions in a problem class have so much structure that they can be recognized or predicted by human intuition, once this structure is known. One example is a work about designing auxetic structures that can be deployed via inflation or gravity, in which the authors show that feasible shapes are exactly surfaces with positive mean curvature [KLPCP18]. Finally, I want to point to a work about the optimization of mass properties, such as the center of mass and the inertia tensor of an object, by adding an appropriately shaped void to the interior of the object [BWBSH14]. The authors tackle this problem with an approach inspired by voxel-based topology optimization, but the results

suggest that the boundary between void and solid is formed by a circular double cone regardless of the shape of the object. It seems probable that there is a structure to these solutions that is yet to be understood.

It would be overly optimistic to conclude from this short list of examples that the majority of problems in fabrication-aware shape optimization have a beautiful, yet undiscovered solution that makes iterative optimization obsolete. Yet, there are reasons to believe that "inverse problems" are sometimes easier than the computationally intensive "forward simulation" problems with which they are associated, and admit solutions that can be characterized in intuitive terms or computed with stronger guarantees of success.

An obvious argument in favor of this hypothesis is that non-linear PDEs are often linear in the quantities that are assumed unknown in the inverse problem, such as material densities and stiffnesses. However, even certain non-linearities, such as the ones discussed in Appendix A.5 may present no extra complications. Adding to the difficulty of inverse problems instead is that the unknowns are usually subject to inequality constraints, such as the positivity of scalar stiffnesses, as in Section 4.3, positive-definiteness of stiffness tensors, or more complex constraints, such as the ones found in Section 5.4.

A major result of this thesis—which I hope will pleasantly surprise the reader, the way it surprised me—is that the inverse design problems studied here admit solutions that are computationally much simpler than one would suspect in the face of constraints such as these. Another surprising finding is that the inverse problems we solve in Chapters 4 and 5, namely to characterize the equilibrium states of plane elastic curves and Kirchhoff rods with variable stiffness, prove to have close ties to projective line geometry.

I think that an experience shared between many academics at some point during their career is a disappointment with their own work or even the research field that they have chosen. Many of my peers during my doctoral studies happen to work either in analysis or computer graphics, and I have—without any claim to universality—observed that we are disappointed by different things: Analysts are glad that their work is beautiful, rigorous, and eternal, but disenchanted with the suspicion that no real-world problem will ever satisfy the assumptions needed to apply it. Researchers working in computer graphics are proud that "their stuff actually works", but cannot shake the feeling that their research is shallow when compared to sciences that pursue universal truths directly.

A common ground between these groups is the fear that problems admit either solutions that are beautiful, or solutions that are useful, but not both.

My own phase of disappointment began after the feeling of giddiness about the coolness of graphics research had abated, and when I started to think that problems in fabrication-aware optimization are too difficult and too close to real life to say anything beautiful about them. It ended when I recognized that the dichotomy of beauty and utility is false, in part through pursuing this thesis, and by reading some of the work cited here. I hope that this text can contribute a small part to instilling the same sense in the reader.

1.1 Overview

This thesis is compiled of two lines of research that contribute to the field of fabrication-aware inverse shape optimization. The first explores the design space of elastic curves in the plane, and its three-dimensional generalization, Kirchhoff rods. It characterizes exactly the kinds of deformed shapes that straight elastic rods may form at equilibrium, given that we can freely choose their stiffness distribution. The second tackles the problem of solving PDEs on CAD models in such a way that the solution depends differentiably on the CAD parameters. This allows us to perform PDE-constrained shape optimization directly on CAD models while sidestepping the need to convert it to a polygonal representation.

The Design Space of Elastic Rods. This line of work studies a specific inverse shape design problem related to the deformation of thin elastic rods. This problem is inverse in the mathematical sense [Kir11] to that of finding an equilibrium state of an elastic rod whose undeformed geometry—given by its thickness, or more generally, its cross-sectional distribution—and boundary conditions are known. In the problem we study, we assume that we know the equilibrium state (and thus the boundary values), and infer the undeformed geometry.

Chapter 4 is dedicated to the plane problem [HB21], in which the deformed state of an elastic rod is represented by a plane curve, and the stiffness is captured by a single positive scalar at each point of the curve. We show that a curve admits a solution to the inverse problem if and only if there exists a line that intersects the curve exactly in its inflection points. Thus, every non-inflectional curve admits a solution, but for curves having at least one inflection, it depends on its global shape. The structure of all solutions, i.e., all stiffness distributions that equilibrate a given curve, is also determined by the family of (projective) lines having said property, and is isomorphic to it. To make this result useful for fabrication, we pose an optimization problem for finding the stiffness distribution with the highest uniformity, and show that it can be determined by solving a linear program.

To enable real-world applications on a larger scale, we extend this method computationally in two ways. First, we model elastic rods in a gravitational field, and show that the same optimization problem of finding the stiffness distribution with the highest uniformity is still linear, and can be solved by a two-stage linear program. Second, we analyze the stability of multi-inflectional equilibrium curves by using second-order variational methods, such as the ones described in Section 3.3. Unstable equilibria do not maintain their shape under perturbations, so they are unsuited to most applications. Thus, we formulate the problem to find a *similar* equilibrium curve which is stable, and present an algorithm to solve it based on applying the adjoint method to isoperimetric conjugate points of the unstable solution.

Chapter 5 analyzes the inverse problem associated with the deformation of thin elastic rods in three-dimensional space, based on the Kirchhoff rod model [HB22]. To account for both bending and twisting, a deformed state is represented by a framed curve, in which the twist of the moving frame mirrors the physical twist of the rod. There is significantly more design freedom in this problem, because we could potentially pick a differently shaped cross section at every point of the rod in order to match a prescribed equilibrium state. However, one of our results shows that rods composed entirely of elliptical cross sections solve every inverse problem that admits a solution, and therefore it is not necessary to consider more exotic shapes.

We proceed to characterize the inverse problems that have a solution, i.e., the (framed) curves that can be interpreted as equilibria of Kirchhoff rods by picking a suitable distribution of cross sections. While our result is less intuitive than the one in the plane case, it can be stated concisely using objects from projective line geometry, and results in a condition that can be verified and optimized for with a linear program.

The shape optimization algorithms developed in both chapters are implemented in an interactive design system that allows the user to modify curves by dragging spline control points, defining curves analytically, or creating a number of curves by intersecting a surface with a family of planes. The application automatically checks if a curve admits a solution to the inverse problem, and generates the geometry of an elastic rod that realizes this curve as an equilibrium state if possible. All computations involved take only a fraction of a second, so the results can be inspected in real time during user edits.

We used this system to design bending-active models, consisting of parts that can be fabricated in a flat state, and attached to a support structure to take their curved shape at equilibrium. To showcase results of the plane theory, we used paper, cardboard, polyacetal sheets, and thin laminated timber plates as base materials and built models on a scale up of up to 90 cm in length. Results of the three-dimensional theory are manufactured by silicone casting and include a free-form light sculpture and a soft gripper actuated by torsion.

X-CAD. The earlier work [HSK+19], which is presented in Chapter 6, tackles the problem of general-purpose gradient-based optimization of CAD models. CAD design proceeds by building up a three-dimensional model, such as a volume or surface, from parameters and constraints by a linear sequence of operations such the generation of primitive shapes; path extrusion, revolution, and sweeping; adding fillets and chamfers to existing shapes. On a technical level, the result is a boundary representation of the model, consisting of parametric surfaces that are created by the CAD application, but this representation is usually hidden from the user. In order to analyze a CAD model, such as computing its mass properties, analyzing its response to forces, or solving CFD (computational fluid dynamics), the model is usually discretized as a polygonal or volumetric mesh in order to interface with simulation software.

This presents a technical challenge when we attempt to modify the CAD model based on the analysis result in an automated manner. Adjoint methods, as described in Section 3.4, enable us to compute the derivative of the analysis result with respect to input data, such as the vertices of a volumetric mesh, and one could further relate these vertices to the original CAD parameters—albeit with some difficulty, because this functionality is not provided by the commercial CAD kernels currently in use. An insurmountable challenge arises in this workflow once we try to embed it in a gradient-based optimization algorithm such as steepest descent, as described in Section 3.5. For an algorithm such as this one to make progress, the objective function needs to be differentiable at least, but the discretization step is not even continuous because the combinatorics of the mesh may change after modifying the CAD model.

In Chapter 6, we present a solution to this conundrum by not discretizing the CAD model itself, but discretizing a portion of the embedding space that contains the model instead. This discretization is not directly dependent on the geometry of the CAD model, and thus circumvents the issue. However, we are facing new difficulties rooted in the fact that our simulation domain—the interior of the CAD model—retains its curved boundary, which is described by parametric surfaces. In order to apply a finite-element type procedure, we need to intersect this simulation domain with the discretization of the embedding space, which forms elements that are bounded by pieces of these parametric surfaces. Our core

contribution to enable simulation methods in this setting is a way of performing high-accuracy numerical integration on domains bounded by parametric surfaces. The integration result is differentiable in the surface parameters, so it preserves the regularity needed to apply gradient-based optimization.

We demonstrate the efficacy and robustness of the algorithm by performing shape optimization on a range of CAD models with hundreds of parametric surface patches each, and for a variety of objectives. Among them are the optimization of integral quantities such as the center of mass, and performance indicators derived from a linearly elastic simulation such as total compliance and maximal stress. We also perform shape matching between the displacement field resulting from a non-linearly elastic simulation and a target surface, as well as a form of topology optimization, in which CAD parameters are used to emulate topology changes by shrinking and expanding holes in an object.

CHAPTER 2

Related Work

2.1 Fabrication-Aware Shape Optimization

A relatively recent subfield of computer graphics, fabrication-aware shape optimization started to emerge around 2010, fueled by the additive-manufacturing revolution that makes it possible to fabricate objects with high geometric complexity at low cost. Pioneering works from this period tackle the design of elastic 3d-printable materials [BBO+10], the design of 3d-printable interlocking puzzles [LFL09], and paneling problems found in free-form architecture [EKS+10].

The field has since branched in many directions [BFR17], such as designing appearance [ESZ+17] and stylization [DLL+15] of fabricable artifacts; optimizing different aspects of elastic objects such as structural strength [ZPZ13], controlling the deformation under external loads [STC+13], and designing microstructures [PZM+15] with controllable behavior; designing objects for different functionalities, such as spinning tops [BWBSH14], paper planes [UKSI14], and sound filters [LLMZ16]; architecturally-inspired applications such as pop-up models [LJGH11], self-supporting surfaces [VHWP12], flat-to-curved manufacturing [KCD+16] and many more. Much research has also been conducted to optimize the fabrication process itself, by minimizing the material used [WWY+13], optimizing tool paths [ZGH+16], partitioning objects to fit individual parts on the build plate of a 3d-printer [SDW+16], or optimizing support structures for single-material printers [DHL14].

To contain the scope of this chapter, we will focus on applications that target the optimization of elastic objects, especially sheets, rods, and grids. There are a number of articles that the reader can consult for a broader overview, such as a survey of methods and representations used in fabrication-aware design [BFR17], and three more specialized surveys about computational design of rigid-part assemblies [WSP21], on stylized fabrication [BCMP18], and on fabrication-assisted data visualization [DSMA+21].

2.1.1 Elastic Volumes and Sheets

The inverse design of free-form objects from elastic materials has received wide attention in the past ten years and covers a range of applications, such as inverse shape design under gravity [STBG12, CZXZ14], design of objects with prescribed natural frequencies [BLT+15], and design of curved surfaces with elastic membranes [POT17, GMB17], for example by combining a pre-stretched sheet with rods arranged in tiled star patterns [JSVB20]. Guseinov

et al. [GMP+20] propose an inverse design tool to encode spatial and temporal morphing of initially flat, self-actuating shells by controlling the softening rate through mesostructure geometry. Zheng et al. [ZFF+19] optimize for the flat initial configuration of a target structure composed of elastic ribbons using a combination of finite element analysis and a genetic algorithm.

Other strategies for surface fabrication include modeling the deformation of wire mesh with Chebyshev nets [SFCBCV19], approximation with Origami folding [DVTM16], inflation of air channel networks [PIC+21], curved developable surfaces [IRHSH20, SGC18], auxetic shells [KLPCP18], wire meshes mapped to Chebyshev nets [GSFD+14], or packable spirals that can be pulled apart [WPGSH18]. Another line of research focuses on optimizing the strength-to-weight ratio of printed parts [SVB+12, ZKWG16], or on a general-purpose parametrization for optimizing different properties [MAB+15, MHR+16].

2.1.2 Rods and Rod Networks

Solutions to design problems are often based on forward simulation methods, which have been studied extensively. The earliest rod simulation method in graphics literature treats the static clamped-free case [Pai02], while the first dynamic rod simulation method is based on a piecewise-helical discretization [BAC+06]. The popular discrete elastic rod model supports arbitrary boundary conditions, constraints, and anisotropic cross sections [BWR+08, BAV+10].

Rod simulation is frequently used to compute the dynamics of hair in computer animation. Modeling hair styles that look lifelike under gravity is among the earliest inverse design problems studied in computer graphics [Had06], with later works taking into account collisions and frictional contact [DJBDDT13]. In computational fabrication, rod models have been employed to predict the elastic response of wires for instance. Applications include the design of structurally stable wire sculptures [MLB16] and cable-actuated wire characters with multiple target poses [XKCB19], as well as optimizing the motor trajectories of robotic wire characters to minimize mechanical oscillations [HXK+19]. Introducing an interactive design system, He et al. [HPL+19] transform static shapes into deformable objects by customizing the deformation behavior of helical springs. These methods either rely on plastic wire bending or 3d printing for fabrication. Recently, Duenser et al. [DPTC20] used a rod model to robotically control a hot-wire foam cutter with a flexible elastic cutting wire.

Rods and ribbons also play a significant role in self-formation processes, in which mechanical properties are controlled and exploited to realize a design. Examples are spatially-varying thermal properties to induce controlled curvature in rods with a straight initial state [WTC+19] and ribbons whose natural curvature is modified to control their buckled shape [ZFF+19]. Recently, Liu et al. [LDV20] proposed a method for designing elastic planar sheets that can be bent into axisymmetric 3d structures without gaps. Their approach relies on elastic strips whose thickness and width are tapered. This model is tailored for a specific class of shapes, namely surfaces of revolution with at most one inflection point on the meridian curve.

Grids of elastic rods have been used in an architectural context for the economical construction of curved façades and pavilions. One successful concept is that of a gridshell, an elastic lattice that is assembled in a planar configuration and curved during deployment [LAGK13]. Gridshells have been built based on asymptotic nets [SHSP18], special geodesic nets and sliding joints [PLBM20, PM21], and using deployment simulation with inverse design optimization [PKLI+19]. Another approach for constructing curved surfaces from flat elements at scale is the use of spiraling microstructures [MPI+19]. In addition to geometric and simulation-based

approaches, also experimental, hybrid analogue and digital processes [Sym15], or dedicated data-driven models that map knowledge derived from physical form-finding experiments to shapes [FM11] have been explored.

Outside of architecture, rod networks have also been used to approximate deformable surfaces with multiple target poses at a low fabrication cost [PTC+15] and to cover a surface in a structurally sound, decorative network of user-defined patterns [ZCT16]. A recent application of ribbon networks concerns the design of tri-axially woven free-form surfaces, based on near-geodesic families of curves [VZF+19] or using naturally curved ribbons [RPC+21].

2.1.3 Design Spaces

Some works go beyond solving inverse design problems numerically and also provide formal conditions for the feasibility of design problems. Pioneering this type of contribution in graphics literature, Derouet-Jourdan et al. [DJBDT10] study the problem of determining the natural curvature of an isotropic clamped-free elastic rod in the plane whose deformed shape under gravity is known. In addition to providing a numerical inversion algorithm, the authors give a sufficient stability condition on the stiffness and density of the rod. Bertails-Descoubes et al. [BDJRL18] study the same inversion problem for Kirchhoff rods in three dimensions and show that the natural configuration is uniquely determined by the equilibrium curve up to framing.

Beyond the design of elastic rods, Konaković Luković et al. [KLPCP18] study pneumatically-actuated deployment of auxetic structures and show that this design space is described exactly by surfaces with positive mean curvature almost everywhere. Finally, Wang and Solomon [WS21] represent skinning weights on meshes as solutions to a parametrized family of elliptic partial differential equations, which are shown to possess the same properties as high-quality weights. We use a similar characterization, based on solutions to a parametrized differential equation, to describe twist-free equilibria of elastic rods.

2.2 Euler Elastica

The inverse design problem we study in Chapter 4 is based on the so-called elastica problem, first posed precisely by James Bernoulli in 1691. This problem asks for the shape of a thin, uniform elastic rod of negligible weight attached to the ground at one end, and loaded with a weight on the free end. Partial solutions were found by James and Daniel Bernoulli, but Euler was the first to completely characterize all solutions to the problem and its generalizations for different boundary conditions and loadings [Lev08].

Of mathematical interest to the present day are questions of buckling and stability of elastic curves given different boundary conditions, number of inflections, and stiffness distributions. Most work treats the stability of *constant-stiffness* elastic curves, such as an early proof that all non-inflectional curves with kinematic boundary conditions are stable [Bor06, p. 17], and work on a critical load that causes out-of-plane buckling [Kov69]. The most comprehensive study notes that, for the constant-stiffness case "it can readily be shown that the higher modes are all unstable. [...] The case of variable stiffness is not so easily treated, and we reach no conclusions." [Mad81, p. 52]. Recent work examines inflectional curves with different combinations of boundary conditions [SL10, Bat15], and the stability of a doubly-clamped elastica constrained by a curved surface [CH14]. Only a small number of works treat elastic curves with variable stiffness, for specific load cases and stiffness

profiles [Coş10, LL18]. The main technical tools for analyzing stability are eigenvalue analysis and the Jacobi criterion [GF63, MRM98].

2.3 CAD

Computer-aided design is a modern workflow for creating two-dimensional sketches or three-dimensional geometry, used widely by engineers and designers. The basis for this method is the construction of sketches from basic operations such as the creation of primitives, and defining their positions and relationships by dimensioning and adding constraints. Anticipated by the computer program "Sketchpad" [Sut63], CAD programs on personal computers became available with the release of AutoCAD and the predecessor of PTC Creo in the 80s [Hof05]. The feature paradigm, used by modern CAD software like OnShape, is based on work by Roller [Rol91] and Shah [Sha91], as noted by a recent study on CAD design strategies [CCC16].

CAD software ultimately produces models that are bounded by a network of parametric surfaces and curves, such as ones based on B-splines and NURBS [PT96]. Being a smooth representation, this is convenient for differential-geometric analysis, such as the computation of curvatures. Performing structural analysis or solving other simulation tasks is less straightforward on a parametric model, and usually entails a change of representation.

2.3.1 Analysis of CAD Models

The predominant approach for analyzing CAD models is to first volumetrically mesh the enclosed domain, then discretize PDEs over finite elements [SHD+18, HZG+18]. However, whenever a CAD model parameter is adjusted, the conforming representation has to be refined and remeshed. In addition to computational overhead, changes in the discretization can also lead to popping artifacts and non-smooth objective functions during shape optimization. In the last decade, the field of *IsoGeometric Analysis* (IGA) has arisen. IGA attempts to unify design and analysis by augmenting the 2D surface NURBS with 3D solid NURBS elements. This tight integration between CAD and analysis offers tremendous advantages in shape optimization, but it comes with the challenge of identifying a parametrization of the volumetric domain from its boundary, currently an active area of research [HWH+15, DP17]. Mesh-free simulation methods [NTV92, MKN+04, MKB+10] generate approximations of the deformation field from only a set of points, but these methods face the challenge that accurately representing complex features requires an adequate sampling of a model's boundary. Similar to traditional mesh-based approaches, this raises sampling and segmentation issues [FGBP11]. Trying to maintain the advantages of mesh-free methods while addressing these shortcomings, the enrichment concept of the XFEM enables the introduction of additional degrees of freedom (DOFs) for elements cut by the boundary. This enables reduction of the complexity of the geometric meshing domain [FB10]. The method was initially developed by Belytschko and Black [BB99] for modeling crack growth. In graphics, the method has received attention for simulating the cutting of shells and solids [JK09, KMB+09, KBT17].

In Chapter 6, our strategy is to build on the XFEM method in order to avoid complex meshing and remeshing operations. In doing so, we face the technical challenge of accurately integrating on domains delimited by NURBS surfaces. Safardi et al. [SNSG15, SNSG16] suggest to utilize NURBS to augment the finite element approximation space and minimize geometric errors associated with the discretization of a complex domain in combination with the generalized finite element method. Haasemann et al. [HKPU11] developed a quadratic finite element

formulation based on the XFEM and NURBS, which was later extended to higher order approximations [Leg13]. In these works, it is required that the boundary of a model cuts an element edge of the nonconforming mesh at most once, and elements are cut into at most two parts. To fulfill these requirements, elements are recursively subdivided, resulting in discontinuous changes of element domains during shape optimization. A conceptually similar strategy, which also faces similar challenges, is to simplify the integration domain by subdividing elements that are cut by boundaries into curved quadrilaterals and triangles [KZB+15]. However, so far this method has been demonstrated for 2D domains only, and a robust extension to 3D is challenging because of non-trivial configurations emerging from boundaries with complex subelement detail. Alternatively, Müller et al. [MKO13, MKEKO17] showed how to integrate over volumes and surfaces defined by implicitly-given level sets, but their integration strategy may fail when surface features smaller than a simulation element are present. In this thesis, we propose quadrature rules that solve this problem.

2.3.2 Optimization of CAD Models

As topology optimization methods do not impose restrictions on attainable shapes [BS99, LHZ+18], the resulting shapes often cannot be directly fabricated and need to be reinterpreted as CAD models. In contrast, shape optimization introduces a limited set of design variables, and the design problem is formulated directly on the parameterized CAD representation. In practice, shape optimization of CAD models is known to be fragile and delicate to use because of different representations in design and analysis. An extensive review of parametric shape optimization techniques can be found in [DP17].

The potential of using XFEM for shape optimization has already been highlighted by Duysinx et al. [DMJF06], combining simple parametric features such as circles, ellipses, and squares with a level set topology optimization using XFEM. Zehnder et al. [ZKBT17] use a similar approach to optimize a set of spherical inclusions. However, these techniques rely on the assumption that boundaries can be well-approximated by functions which are linear on elements. Therefore, they cannot capture features that are common in CAD models.

Recently, Najafi et al. [NSTG17] introduced an optimization scheme built on NURBS-based interface-enriched generalized FEM (IGFEM), illustrating the application for materials with inclusions. While conceptually we share a similar vision, our method differs in several ways: Firstly, the NURBS-enhanced IGFEM handles problems with weak discontinuities, such as different material interfaces; our method can handle strong discontinuities, such as cuts. Secondly, their method is described and was implemented and validated only for 2D shape optimization problems. Finally, an extension of their integration strategy to 3D would require a tessellation of the volume with 3D NURBS elements, resulting in similar challenges as faced by other IGA methods. We demonstrate the applicability of our method to complex 3D CAD models.

Notably, taking a data-driven approach, Schulz et al. [SXZ+17] proposed an interactive design exploration for CAD models in which an analysis is precomputed for parameter samples on an adaptive grid and then interpolated during run-time. Although this approach is promising for shapes with just a few design parameters, the combinatorial complexity becomes prohibitive for higher-dimensional problems.

CHAPTER 3

Technical Preliminaries

3.1 Differential Geometry of Curves

We use parametrized curves from differential geometry as a mathematical model to capture elastic deformations of slender objects such as beams and rods. It can be shown that, under moderate loads, the deformation of objects such as these are well approximated under the simplifying assumptions that the centerline is in a state of pure bending, and the cross sections only transform isometrically. This is the force regime that we are interested in, because it allows the creation of interesting curved shapes while precluding material failure, which would be the result of forces large enough to cause significant levels of tensile strain. Therefore, it suffices to prescribe only the deformed centerline and cross-sectional orientation in order to reconstruct the full state of deformation, and compute quantities such as the elastic energy. This makes plane curves, for in-plane bending, and framed curves in $\mathbb{R}^3$, for rods that undergo bending and twisting, the standard models for capturing these types of deformations.

In this section, we introduce relevant concepts from the differential geometry of plane curves and framed curves embedded in $\mathbb{R}^3$, which will be used as models for plane elastic curves in Chapter 4 and elastic rods undergoing bending and torsion in Chapter 5, respectively. While we explore these concepts, we keep in mind the analogy to elastic rods, and point out how differential-geometric quantities relate to elasticity theory.

The presentation in this section largely follows lecture notes by Hertrich-Jeromin [HJ19]. For readers familiar with the classical theory of Frenet curves, but not with that of general moving frames, Bishop [Bis75] provides a concise introduction. More details are given by O'Neill [O'N66, Ch. 2].

3.1.1 Plane Curves

A continuously differentiable map $\gamma : I \to \mathbb{R}^2$, with $I = (t_0, t_1) \subset \mathbb{R}$ an interval, is called a *regular parametrized curve* if $\forall s \in I : \gamma'(s) \neq 0$. If we regard the parameters $s \in I$ as the material points on the centerline of an elastic body, we can interpret γ as a deformation of the centerline, which moves a point s to $\gamma(s)$.

The regularity of γ tells us something about the types of deformations that are allowed in our model. Continuity of γ guarantees that the deformation does not disconnect (fracture) the centerline. The derivative $\gamma'(s)$ encodes the direction of the centerline at s, as well the the

factor by which the centerline is stretched or compressed, given by $\|\gamma'(s)\|$. By assuming that γ' is continuous and $\gamma' \neq 0$, we exclude the possibility that the stretch factor approaches zero at any point, or even contracts a finite segment of the centerline to a point.

Arc-Length Parametrization. Any map $\tilde{\gamma} = \gamma \circ f : \tilde{I} \to \mathbb{R}^2$ with $f : \tilde{I} \to I$ surjective, continuously differentiable, and $f' > 0$ is called an *orientation-preserving reparametrization* of γ, and is also regular. Among all reparametrizations of γ, there exists a unique choice $\bar{\gamma} : (0, \ell) \to \mathbb{R}^2$ that satisfies $\|\bar{\gamma}'\| \equiv 1$. It holds that $\ell = \int_I \|\gamma'\|$, which is called the *length* of the curve[1]. To show existence, we define

$$g : I \to (0, \ell) : t \mapsto \int_{t_0}^{t} \|\gamma'\|,$$

and set $f = g^{-1}$, which exists because $g' = \|\gamma'\| > 0$. Then, we compute

$$\|\bar{\gamma}'(s)\| = \|\gamma'(g^{-1}(s)) \cdot \frac{1}{g'(g^{-1}(s))}\| = \|\gamma'(g^{-1}(s))\| \cdot \frac{1}{\|\gamma'(g^{-1}(s))\|} = 1.$$

We call $\bar{\gamma}$ the *arc-length reparametrization* of γ.

As discussed in the introductory part of this section, we are ultimately interested in describing deformations in which the centerline undergoes bending but does not stretch. This assumption translates directly into the arc-length condition $\|\gamma'\| \equiv 1$. Working directly with arc-length parametrized curves is convenient mathematically, because it simplifies many formulas. However, the arc-length reparametrization of most curves—exceptions being lines and circles—cannot be expressed in closed form and is therefore computationally unwieldy.

A useful compromise is to view arc-length parametrized curves as the equivalence classes of the reparametrization operator on the set of all regular parametrized curves. This way, we view any regular curve as a *representative* of the arc-length parametrized curve to which it can be reparametrized. This allows one to argue about arc-length parametrized curves but use a representative with a closed-form parametrization for computational purposes. From now on, we will assume that γ is arc-length parametrized to simplify the exposition. Whenever it is relevant to a computer implementation of the algorithms discusses in this work, we will also note which formula needs to be used when an arc-length parametrization is not available.

Turning Angle. Assuming $\|\gamma'\| \equiv 1$, there exists $\alpha : (0, \ell) \to \mathbb{R}$ such that $\gamma' = (\cos\alpha, \sin\alpha)^t$, where $(\cdot)^t$ denotes matrix (or vector) transposition. Because γ' is continuous, we can also choose α to be continuous. If, in addition, we choose $\alpha(0) \in [0, 2\pi)$, this uniquely defines the *turning angle* α. After choosing a starting point $\gamma(0)$, the reverse mapping from α to γ is also unique: $\gamma(s) = \gamma(0) + \int_0^s (\cos\alpha, \sin\alpha)^t$.

Curvature. The quantities introduced so far, namely γ, γ', and α, change if a curve is subjected to a rigid transformation, i.e., rotated and translated in the plane as a whole. Naturally, rigid transformations of an elastic object do not change its elastic energy, and thus it is easiest to describe the elastic energy in terms of quantities that are *invariant* under these transformations—so-called *geometric invariants*.

[1]The integral of a function $f : I \to \mathbb{R}$ will usually be denoted by $\int_I f$ following Lieb and Loss [LL01, Ch. 1.1], without introducing the variable of integration or the integration measure explicitly—the latter will always be the Lebesgue measure. In case the variable of integration needs to be named, for example to compute the first moment of a function, we employ the usual notation $\int_I x f(x)\,\mathrm{d}x$.

For plane curves, the most fundamental geometric invariant is the *(signed) curvature* $\kappa = \alpha' : (0, \ell) \to \mathbb{R}$, whose definition requires γ to be twice-differentiable. For practical purposes, we will even require γ to be twice continuously-differentiable, so κ is a continuous function. The reverse mapping from κ to α is unique once we fix $\alpha(0)$, via $\alpha(s) = \alpha(0) + \int_0^s \kappa$. The mapping from κ to γ is unique after fixing both $\gamma(0)$ and $\alpha(0)$, and all curves with the same signed curvature are exactly all possible rigid transformations of each other.

To express κ directly in terms of γ, we compute $\gamma'' = (-\alpha' \sin\alpha, \alpha' \cos\alpha)^t = \mathcal{J}\gamma'\kappa$, where $\mathcal{J} = \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$ represents the counter-clockwise rotation by $90°$ in the plane. This shows that $\langle \gamma', \gamma'' \rangle = \kappa \langle \gamma', \mathcal{J}\gamma' \rangle \equiv 0$, and

$$\det(\gamma', \gamma'') = \det(\gamma', \mathcal{J}\gamma'\kappa) = \kappa \langle \mathcal{J}\gamma', \mathcal{J}\gamma' \rangle = \kappa,$$

where we used the identity $\det(v, w) = \langle \mathcal{J}v, w \rangle$, which holds for any $v, w \in \mathbb{R}^2$. To compute κ directly from a representative $\tilde{\gamma} = \gamma \circ f$ which is not arc-length parametrized, we first compute

$$\tilde{\gamma}'(t) = \gamma'(s) \cdot f'(t), \quad \tilde{\gamma}''(t) = \gamma''(s) \cdot [f'(t)]^2 + \gamma'(s) \cdot f''(t),$$

where we have abbreviated $s = f(t)$. Next we investigate what happens if we apply the determinant formula to $\tilde{\gamma}$ instead of γ and find

$$\det(\tilde{\gamma}', \tilde{\gamma}'') = \langle \mathcal{J}\tilde{\gamma}', \tilde{\gamma}'' \rangle = \langle \mathcal{J}\gamma' f', \mathcal{J}\gamma'\kappa(f')^2 + \gamma' f'' \rangle = \kappa (f')^3,$$

so we have to correct this formula through division by $(f')^3 = \|\tilde{\gamma}'\|^3 > 0$, yielding

$$\kappa = \frac{\det(\tilde{\gamma}', \tilde{\gamma}'')}{\|\tilde{\gamma}'\|^3}.$$

The curvature at a point s is a measure of how strongly the curve bends, and in which direction it bends, encoded by the magnitude and sign of $\kappa(s)$, respectively. The curves for which κ vanishes identically are exactly the line segments, and the curves for which $\kappa \equiv c \neq 0$ are exactly the circle arcs, in which case $1/c = \pm r$, with r the radius of the circle.

In Chapter 4, the points at which κ vanishes will play an important role. These are called the *inflection points* of γ, and they indicate the points at which the direction of bending changes from clockwise to counter-clockwise, or vice versa.

Elastic Energy. An *elastic potential energy* $U(\gamma)$ is a measure of how much work is necessary to induce a certain deformation γ, so it is usually easiest to express in terms of κ for its geometric invariance. If κ is identically zero, γ is undeformed, so its elastic potential vanishes, and it increases with the magnitude of κ. Assuming that there exists an *energy density* W which measures the local contribution to U at each s, i.e., $U = \int_0^\ell W(s, \kappa(s))\,\mathrm{d}s$, we have $W(s, 0) = 0$ for all $s \in (0, \ell)$. This expresses that straight segments of a curve do not carry an elastic potential. Furthermore, one usually has $W(s, \kappa) = W(s, -\kappa)$, which indicates that bending a slender object clockwise takes the same amount of work as bending it counter-clockwise.

The simplest and most common elastic energy density of this type is given by $W(s, \kappa(s)) = \frac{1}{2} K(s) \kappa^2(s)$ where $K : (0, \ell) \to \mathbb{R}_{>0}$ is called the *stiffness function*. Regions of the curve for which the stiffness is high contribute more strongly to the elastic energy than regions for which the stiffness is low. This is used to model, for example, non-uniform beams, which may be wider or thicker in certain regions than in others. Another common assumption is

to strengthen the condition $K > 0$, and to require that there exists a constant $c > 0$ such that $K(s) \geq c$ for all $s \in (0, \ell)$. Such stiffness functions are called *coercive*, and they simplify the analysis of curves which minimize the elastic energy—a subject we will come back to in Section 3.3.

3.1.2 Framed Curves in $\mathbb{R}^3$

The concept of a parametrized plane curve can be generalized to an ambient space with a higher dimension as $\gamma : (0, \ell) \to \mathbb{R}^d$ continuously differentiable with $d \geq 2$. Some definitions carry over directly from the plane case: γ is called *regular* if $\|\gamma'(t)\| > 0$ for all t, and *parametrized by arc length* if $\|\gamma'\| \equiv 1$. The orientation-preserving arc-length reparametrization of a curve is also unique for arbitrary d.

The analogy with elastic rods still makes sense for $d = 3$, which is the case that we will consider here. Similar to the plane case, we can view γ as a deformation of the centerline of a rod, which moves a point $s \in (0, \ell)$ to $\gamma(s)$. Then $\|\gamma'(s)\|$ is a measure of the local stretching or compression of the centerline. But since we are only interested in the force regime that leaves the centerline unstretched, we will again view a curve as a representative of its arc-length reparametrization.

The definition of signed curvature does not carry over directly from $d = 2$ to $d = 3$. For plane curves, the sign encodes whether the curve winds clockwise or counterclockwise—but this property changes depending on whether we view the curve from the front or from the back. For a curve embedded in $\mathbb{R}^3$, there is no canonical viewing direction, so only the magnitude of the curvature carries meaningful information. In analogy to the determinant formula in $\mathbb{R}^2$, we define the *(geometric) curvature* $\kappa = \|\gamma' \times \gamma''\|$, which coincides in magnitude with the signed curvature of the plane curve obtained by orthogonally projecting γ onto its best-approximating plane at a point.

A concept that cannot be represented by a parametrized curve in $\mathbb{R}^3$ alone is the twist of an elastic rod, or the rotation of cross sections around the centerline. To represent twist, we need to add more information to a parametrized curve, and this is usually done with a moving frame.

Moving Frame. A pair $(\gamma, F) : (0, \ell) \to \mathbb{R}^d \times SO(d)$ with $\gamma' = F \cdot e_d$ and F continuously differentiable is called a *framed curve*, and F a *moving frame* adapted to γ. Here, e_i denotes the i-th standard basis vector of $\mathbb{R}^d$. For $d = 2$, γ uniquely defines F, so the moving frame carries no additional information. For $d = 3$, the manifold of all moving frames adapted to a fixed curve is diffeomorphic to the space of differentiable functions from $(0, \ell)$ to S^1, so we can choose a rotation angle around the centerline at each point.

To see this, write the columns of F as (n_1, n_2, γ'), where n_1 and n_2 are called *material normals*, and we have $n_2 = \gamma' \times n_1$. Given a second moving frame $\bar{F}$ adapted to γ', we write its columns as $(\bar{n}_1, \bar{n}_2, \gamma')$, with $\bar{n}_2 = \gamma' \times \bar{n}_1$. Because $\langle n_1, \gamma' \rangle \equiv 0 \equiv \langle \bar{n}_1, \gamma' \rangle$, there is a unique $\beta : (0, \ell) \to S^1$ such that

$$\bar{F} = (\bar{n}_1, \bar{n}_2, \gamma') = (n_1, n_2, \gamma') \begin{pmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{pmatrix} = FQ_\beta.$$

This shows that F encodes a planar rotation, i.e., a rotation in the normal plane spanned by n_1 and n_2, at each point of the curve.

A moving frame can be used to encode the twist of an elastic rod by viewing n_1 and n_2 as coordinate axes of a cross section, emanating from the intersection point of the centerline and a cross-sectional plane. This way, if a cross section rotates around the centerline as a result of twist, this can be captured by applying the same rotation to n_1 and n_2.

Material Curvatures and Twist. A moving frame and its derivatives are not geometrically invariant because they change when the framed curve undergoes a rigid transformation. To find a geometric invariant related to F, we make use of its orthogonal structure and differentiate the equation $FF^t \equiv \mathrm{id}$, which yields $F'F^t + F(F')^t \equiv 0$. We can rewrite this equation as $F'F^t = -(F'F^t)^t$, which shows that $F'F^t$ is a skew-symmetric matrix. Skew-symmetric 3-by-3 matrices have the property that their action equals that of a cross product, i.e., we can find $\omega \in \mathbb{R}^3$, such that $F'F^t \cdot v = \omega \times v$ for all $v \in \mathbb{R}^3$. To express this relationship, we also write $[\omega]_\times = F'F^t$. The map $\omega : (0, \ell) \to \mathbb{R}^3$ is called the *Darboux vector* of F.

The Darboux vector itself is not geometrically invariant, but its coordinates k with respect to F are. These coordinates, given by $k = F^t\omega$, are called the *curvature vector* of F. To show this invariance, we make use of a short lemma: Let $Q \in SO(3)$ and $v \in \mathbb{R}^3$. Then, $Q[v]_\times Q^t = [Qv]_\times$. To see that this holds, verify that

$$Q[v]_\times Q^t w = Q(v \times (Q^t w)) = (Qv) \times w = [Qv]_\times w$$

for any $w \in \mathbb{R}^3$. Given a rotated copy of the framed curve, $(\bar\gamma, \bar F) = (Q\gamma, QF)$, we compute $[Q\omega]_\times = Q[\omega]_\times Q^t = QF'F^tQ^t = \bar F'\bar F^t$, so its Darboux vector is given by $\bar\omega = Q\omega$. Then we see that its curvature vector is $\bar k = \bar F^t\bar\omega = F^tQ^tQ\omega = F^t\omega = k$.

To interpret the curvature vector geometrically, it is instructive to write it purely in terms of F. We do this by computing $[k]_\times = [F^t\omega]_\times = F^t[\omega]_\times F = F^tF'F^tF = F^tF'$, or $F' = F[k]_\times$. This equation is called the *structure equation* of a framed curve, and it can be expanded as

$$\begin{pmatrix} | & | & | \\ n_1' & n_2' & \gamma'' \\ | & | & | \end{pmatrix} = \begin{pmatrix} | & | & | \\ n_1 & n_2 & \gamma' \\ | & | & | \end{pmatrix} \cdot \begin{pmatrix} 0 & -\tau & \kappa_2 \\ \tau & 0 & -\kappa_1 \\ -\kappa_2 & \kappa_1 & 0 \end{pmatrix}. \tag{3.1}$$

This shows that the components of k give the velocities of the columns of F in terms of the columns of F themselves. The components $k = (\kappa_1, \kappa_2, \tau)^t$ can be extracted to give

$$\begin{aligned} \kappa_1 &= \langle n_2', \gamma' \rangle = -\langle \gamma'', n_2 \rangle, \\ \kappa_2 &= \langle \gamma'', n_1 \rangle = -\langle n_1', \gamma' \rangle, \\ \tau &= \langle n_1', n_2 \rangle = -\langle n_2', n_1 \rangle. \end{aligned}$$

The *twist* τ measures how fast n_1 and n_2 rotate around the tangent, and thus coincides with our intuitive notion of twisting a cable. The *material curvatures* κ_1 and κ_2 measure, up to sign, how fast the tangent vector γ' rotates in directions n_1 and n_2, or in other words, how much the curve bends as we walk along it.

The material curvatures are related to, but contain more information than the geometric curvature κ of γ. They encode how much of the curvature is due to rotation around n_1 and n_2 each, which becomes important if the cross section of an elastic rod is anisotropic. In this case, the resistance to bending may differ depending on the direction: Imagine trying to bend a paper ribbon in such a way that the surface of the ribbon stays in-plane. One would be met with enough resistance to rip the paper, while bending it in the "natural" way needs very little force. This phenomenon makes it important to keep track of κ_1 and κ_2 separately.

Frame Invariants. Even though k and ω depend on the choice of frame, certain dependent quantities are the same across all frames adapted to a fixed curve. These are called *frame invariants*, and they can be computed by relating k and ω to quantities that depend only on γ. To establish a frame invariant involving κ_1 and κ_2, we first compute

$$\gamma' \times \gamma'' = \gamma' \times (\kappa_2 n_1 - \kappa_1 n_2) = \kappa_2 n_2 + \kappa_1 n_1, \tag{3.2}$$

which makes use of Eq. 3.1 and the orthogonal structure of F. Then, we see that

$$\kappa^2 = \|\gamma' \times \gamma''\|^2 = \|\kappa_2 n_2 + \kappa_1 n_1\|^2 = \kappa_1^2 + \kappa_2^2,$$

so the Euclidean norm of $(\kappa_1, \kappa_2)^t$ equals κ and thus only depends on γ, but not on F. Using Eq. 3.2 once more, we also find

$$\gamma' \times \gamma'' = \kappa_1 n_1 + \kappa_2 n_2 = Fk - \tau\gamma' = Fk - \langle Fk, \gamma' \rangle \gamma' = \omega - \langle \omega, \gamma' \rangle \gamma'.$$

This shows that $\gamma' \times \gamma''$ equals the orthogonal projection of the Darboux vector onto the normal plane. Thus, this projection also depends on γ only, and not on F.

Transformations between Frames. We saw at the beginning of this section that any two frames F and $\bar{F}$ adapted to the same curve are related by a rotation $\beta : (0, \ell) \to \mathbb{R}$ in the normal plane, so $\bar{F} = FQ_\beta$. Knowing β, we can compute how the curvature vector and Darboux vector differ between these two frames. To establish these results, we first note that

$$Q_\beta' = \beta' \cdot \begin{pmatrix} -\sin\beta & -\cos\beta & 0 \\ \cos\beta & -\sin\beta & 0 \\ 0 & 0 & 0 \end{pmatrix} = \beta' \cdot \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{pmatrix} = \beta' [e_3]_\times Q_\beta.$$

We use this identity to compute the Darboux vector $\bar{\omega}$ of $\bar{F}$ by evaluating

$$\begin{aligned} [\bar{\omega}]_\times &= \bar{F}' \bar{F}^t = (F'Q_\beta + \beta' F [e_3]_\times Q_\beta) Q_\beta^t F^t \\ &= F'F^t + \beta' F [e_3]_\times F^t = [\omega + \beta' F e_3]_\times = [\omega + \beta'\gamma']_\times, \end{aligned}$$

which makes use of the linearity of the $[\cdot]_\times$ operator. As a consequence, we have $\bar{\omega} = \omega + \beta'\gamma'$, which confirms our result that the Darboux vectors of different frames adapted to the same curve only differ by their tangential component.

A similar computation establishes the connection between the curvature vectors:

$$\begin{aligned} [\bar{k}]_\times &= \bar{F}^t \bar{F}' = Q_\beta^t F^t (F'Q_\beta + \beta' F [e_3]_\times Q_\beta) = Q_\beta^t (F^t F' + \beta' [e_3]_\times) Q_\beta \\ &= Q_\beta^t [k + \beta' e_3]_\times Q_\beta = [Q_\beta^t k + \beta' Q_\beta^t e_3]_\times = [Q_\beta^t k + \beta' e_3]_\times, \end{aligned}$$

so $\bar{k} = Q_\beta^t k + \beta' e_3$, or written in components,

$$\begin{pmatrix} \bar{\kappa}_1 \\ \bar{\kappa}_2 \end{pmatrix} = \begin{pmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{pmatrix} \begin{pmatrix} \kappa_1 \\ \kappa_2 \end{pmatrix}, \quad \bar{\tau} = \tau + \beta'.$$

The formula for the material curvatures expresses that the normal components of ω and $\bar{\omega}$ are equal, but that their coordinates k and $\bar{k}$ are computed once with respect to F and once with respect to $\bar{F}$. Thus $(\bar{\kappa}_1, \bar{\kappa}_2)^t$ is related to $(\kappa_1, \kappa_2)^t$ by a coordinate change in the normal plane furnished through a rotation by β. On the other hand, the twist depends not on the absolute rotation β but on its rate of change. Intuitively, this corresponds to holding a cable between two hands and rotating only hand to introduce twist—rotating both hands would rotate the cable as a whole, but not twist it.

3.2 The Implicit Function Theorem

Introduction. If two variables $x \in \mathbb{R}^n$ and $p \in \mathbb{R}^m$ are related by an explicit functional relationship $g : \mathbb{R}^m \to \mathbb{R}^n : p \mapsto x$, with g differentiable, then the Jacobian matrix $J_g : \mathbb{R}^m \to \mathbb{R}^{n\times m}$ is the best local linear approximation of g around a fixed point $p_0 \in \mathbb{R}^m$. This is expressed by

$$x = g(p) = g(p_0) + J_g(p_0) \cdot (p - p_0) + r(p - p_0),$$

where the first two summands comprise an affine function approximating g around p_0, and $r : \mathbb{R}^m \to \mathbb{R}^n$ is a residual error function such that $r(h)$ goes to zero faster than linearly as $h \to 0$. This is formally captured by the condition $\lim_{h\to 0} \|r(h)\|/\|h\| = 0$. The entries of the Jacobian can be shown to equal the partial derivatives

$$J_g(p) = \begin{pmatrix} \frac{\partial g_1}{\partial p_1}(p) & \cdots & \frac{\partial g_1}{\partial p_m}(p) \\ \vdots & \ddots & \vdots \\ \frac{\partial g_n}{\partial p_1}(p) & \cdots & \frac{\partial g_n}{\partial p_m}(p) \end{pmatrix},$$

where $p = (p_1, \ldots, p_m)^t$ and $g(p) = (g_1(p), \ldots, g_n(p))^t$. Being able to numerically compute a local linear approximation around an arbitrary point p_0 is indispensable for implementing root-finding algorithms such as Newton's method, or local optimization algorithms such as the method of gradient descent.

Sometimes, there exists a functional relationship $g(p) = x$ in a mathematical sense, but it is not available in closed form. Instead, we may have access to an *implicit* relationship of the form $f(p, x) = 0$, where $f : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$. As an example, consider $n = 1 = m$ and $f(p, x) = xe^x - p$. For $p > 0$, there is a unique $x > 0$ such that $xe^x - p = 0$, so there exists a function $g : \mathbb{R}_{>0} \to \mathbb{R}_{>0}$ such that $f(p, g(p)) = 0$ for all $p > 0$. However, one can show that g cannot be expressed in closed form, so we need to resort to numerical methods to compute $g(p_0)$ for a given $p_0 > 0$ by finding a root of $x \mapsto f(p_0, x)$ instead.

Going back to the general case, we have $x \in \mathbb{R}^n$ and $f(p, x) \in \mathbb{R}^n$. If we fix $p_0 \in \mathbb{R}^m$, the map $h : x \mapsto f(p_0, x) : \mathbb{R}^n \to \mathbb{R}^n$ has the chance of being invertible because its codomain and domain have the same dimension. If it really is invertible, then $h^{-1}(0) = x_0$ such that $f(p_0, x_0) = 0$. Via this construction, we can assign to every $p_0 \in \mathbb{R}^m$ exactly one $x_0 \in \mathbb{R}^n$ such that $f(p_0, x_0) = 0$ is satisfied—meaning that we have mathematically constructed an explicit relationship $g : \mathbb{R}^m \to \mathbb{R}^n : p \mapsto x$ from an implicit one, but it may fail to have a closed-form expression in general.

The implicit function theorem. This abstract construction of g only works if h is indeed invertible for all $p_0 \in \mathbb{R}^m$. The implicit function theorem provides a set of sufficient conditions on f that guarantee that this is the case. What makes this theorem very useful in practice is that the Jacobian of g is shown to possess a closed-form expression even when g does not. This lets us use an implicit functional relationship for root-finding and local optimization algorithms in much the same way as an explicit one.

We can write the Jacobian matrix of f in block form as $J_f(p, x) = \begin{pmatrix} J_{f,p}(p, x) & J_{f,x}(p, x) \end{pmatrix}$, or

$$J_f(p, x) = \begin{pmatrix} \frac{\partial f_1}{\partial p_1}(p, x) & \cdots & \frac{\partial f_1}{\partial p_m}(p, x) & \frac{\partial f_1}{\partial x_1}(p, x) & \cdots & \frac{\partial f_1}{\partial x_n}(p, x) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \underbrace{\frac{\partial f_n}{\partial p_1}(p, x) & \cdots & \frac{\partial f_n}{\partial p_m}(p, x)}_{J_{f,p}(p,x)} & \underbrace{\frac{\partial f_n}{\partial x_1}(p, x) & \cdots & \frac{\partial f_n}{\partial x_n}(p, x)}_{J_{f,x}(p,x)} \end{pmatrix},$$

where the n-by-m block on the left contains the Jacobian matrix with respect to p, and the n-by-n block on the right, the Jacobian matrix with respect to x. The implicit function theorem states that $g : \mathcal{N}(p_0) \to \mathbb{R}^n$ with $f(p, g(p)) = 0$ exists for some neighborhood $\mathcal{N}(p_0)$ of $p_0 \in \mathbb{R}^m$ if $J_{f,x}(p, x)$, the n-by-n block on the right, is invertible for all (p, x) in some neighborhood of (p_0, x_0), where $f(p_0, x_0) = 0$. Instead of a rigorous proof, we will give a heuristic argument that this is true, and that we can compute the Jacobian of g in closed form.

For a fixed $p_0 \in \mathbb{R}^m$, the block $J_{f,x}(p_0, x)$ is also the Jacobian of $h : x \mapsto f(p_0, x) : \mathbb{R}^n \to \mathbb{R}^n$. If this matrix is invertible around some $x_0 \in \mathbb{R}^n$ with $f(p_0, x_0) = 0$, then h is also invertible at x_0 because the linear approximation of h at x_0, given by $x \mapsto J_{f,x}(p_0, x_0) \cdot (x - x_0)$ is invertible. This lets us define $g(p_0) := h^{-1}(0)$ uniquely.

Without proving the differentiability of g rigorously—this is done by Marsden et al. [MRA88, 2.5]—we can formally compute its Jacobian. Knowing $f(p, g(p)) = 0$, we differentiate with respect to p, which yields

$$J_{f,p}(p, g(p)) + J_{f,x}(p, g(p)) \cdot J_g(p) = 0$$

by using the chain rule from multivariable calculus. Under the assumption that $J_{f,x}$ is invertible, we can compute J_g as

$$J_g(p) = -[J_{f,x}(p, g(p))]^{-1} \cdot J_{f,p}(p, g(p)). \tag{3.3}$$

Going back to the example $f(p, x) = xe^x - p$, this lets us compute $g'(p)$—which is often written as $\frac{\mathrm{d}x}{\mathrm{d}p}$ to emphasize that we are computing the linearization of x as a function of p. In this example, $J_{f,x}$ and $J_{f,p}$ are scalars given by $J_{f,x}(p, x) = (x + 1)e^x$ and $J_{f,p}(p, x) = -1$. Using Eq. 3.3 yields $\frac{\mathrm{d}x}{\mathrm{d}p}(p) = \frac{1}{(x+1)e^x}$, where x on the right-hand side is such that $p = xe^x$. Note that for this one-dimensional example, we could have gotten the same result by defining the function $p(x) = xe^x$, and computing the derivative of the inverse using the inverse function rule $(p^{-1})'(y) = \frac{1}{p'(p^{-1}(y))}$. But for $n > 1$, the implicit function theorem becomes an indispensable tool, as illustrated in a concluding example.

Example. We work out an application of the implicit function theorem, a variation of which is used in Section 6.5.3. Assume we are given two parametrized surfaces $\sigma_1, \sigma_2 : \mathbb{R}^2 \to \mathbb{R}^3$ that intersect along a curve $\mathcal{C}$ in $\mathbb{R}^3$. In addition, we are given a family of planes $P(d) = \{x \in \mathbb{R}^3 : \langle n, x\rangle - d = 0\}$ with fixed normal vector $n \in S^2$ such that each plane $P(d)$ for $d \in \mathbb{R}$ intersects $\mathcal{C}$ in a single point $q(d) \in \mathbb{R}^3$. Generally, $\mathcal{C}$ will not possess a closed-form parametrization, and $q(d)$ cannot be computed in closed form either. However, after we find $q(d)$ using numerical methods, we can evaluate $q'(d)$—the direction and speed at which the intersection point moves when we change d—analytically thanks to the implicit function theorem.

To do this, we formalize the implicit relationship between the two surfaces and the plane. The distance d takes the place of the variable p from the exposition, because it is the variable with respect to which we want to differentiate. To express that the parametrized surfaces meet the plane in a single point, we introduce surface parameters $u_1, u_2 \in \mathbb{R}^2$, and set $x = \left(\begin{smallmatrix} u_1 \\ u_2 \end{smallmatrix}\right) \in \mathbb{R}^4$. The implicit relationship is then encoded by

$$f(d, x) = \begin{pmatrix} \sigma_1(u_1) - \sigma_2(u_2) \\ \langle n, \sigma_1(u_1)\rangle - d \end{pmatrix} = 0_{4\times 1}$$

We can evaluate the blocks of the Jacobian of f from

$$J_{f,d} = \begin{pmatrix} 0_{3\times 1} \\ -1 \end{pmatrix}, \quad J_{f,x}(x) = \begin{pmatrix} J_{\sigma_1}(u_1) & -J_{\sigma_2}(u_2) \\ n^t J_{\sigma_1}(u_1) & 0_{1\times 2} \end{pmatrix} \in \mathbb{R}^{4\times 4},$$

where $J_{\sigma_1}, J_{\sigma_2} \in \mathbb{R}^{3\times 2}$ and $n^t J_{\sigma_1} \in \mathbb{R}^{1\times 2}$. Then we can compute

$$\begin{pmatrix} \frac{\mathrm{d}u_1}{\mathrm{d}d} \\ \frac{\mathrm{d}u_2}{\mathrm{d}d} \end{pmatrix} = \frac{\mathrm{d}x}{\mathrm{d}d} = -[J_{f,x}(x)]^{-1} J_{f,d}.$$

The derivative of $q(d) = \sigma_1(u_1)$ can then be computed using the chain rule, from

$$q'(d) = J_{\sigma_1}(u_1) \cdot \frac{\mathrm{d}u_1}{\mathrm{d}d}.$$

We will use this theorem at several points throughout this work, for example in Section 4.5 to compute the gradient of an inflection point $s_0 \in I$ of a spline curve $\gamma : I \to \mathbb{R}^2$, in terms of the control points $p_1, \ldots, p_n \in \mathbb{R}^2$ of the curve. Here, an inflection point is defined implicitly via $\det(\gamma', \gamma'')|_{s_0;p_1,\ldots,p_n} = 0$. Another example is found in Chapter 6, where the theorem is used to compute the derivative of intersection points between curves and surfaces, in terms of the parameters defining these curves and surfaces.

3.3 Variational Problems

Variational calculus is a mathematical tool to analyze the stationary points of smooth functionals, i.e., maps from a function space to the reals. There are many classical problems that can be solved with this technique, and what they have in common is that the unknown of the problem is infinite-dimensional, such as a differentiable function $y : (0, \ell) \to \mathbb{R}$.

One of the oldest examples of a variational problem is the brachistochrone problem, posed by John Bernoulli in 1696 [Gol80]. This problem asks for the shape of a slide connecting two fixed points $(0, h) \in \mathbb{R}^2$ and $(w, 0) \in \mathbb{R}^2$ that will take a frictionless ball the least time to roll down under gravity. We can model the shape of the slide as the graph of a function $y : (0, w) \to \mathbb{R}$ such that $y(0) = h$ and $y(w) = 0$. If the ball is modeled as a particle in a gravitational field that starts at rest from $(0, h)$ before rolling down, one can show that the time it takes to reach $(w, 0)$ is

$$F(y) = \frac{1}{\sqrt{2g}} \int_0^w \sqrt{\frac{1 + [y'(x)]^2}{h - y(x)}}\, \mathrm{d}x,$$

where $g > 0$ is the gravitational acceleration. Note that F is a function that maps from a function space to the reals; one calls such functions *functionals*. The goal in this problem is to find, among all differentiable functions satisfying the endpoint boundary conditions, the one that renders F minimal.

The brachistochrone problem is an example of a *simplest* variational problem, called so at least since Bolza [Bol04]. Problems of this type are characterized by the existence of a function $f : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ such that F can be written as $F(y) = \int_0^w f(x, y(x), y'(x))\, \mathrm{d}x$.[2] Many

[2] For the brachistochrone, we have $f(x, y, z) = \frac{1}{\sqrt{2g}} \sqrt{\frac{1+z^2}{h-y}}$.

classical variational problems, such as the Snell's law, geodesic, and minimal surface problems, are of this type and can be studied using the same theory. Note that this problem is called "simplest" because dropping the dependence of f on $y'(x)$ would trivialize the problem, as we could find the optimal $y(x)$ for every fixed $x \in (0, w)$ separately by choosing the minimizer of $y \mapsto f(x, y)$. Problems in which f also depends on $y'(x)$, which couples the different values y obtains, is the simplest *interesting* variational problem that has been studied extensively.

The calculus of variations is a set of techniques for systematically analyzing F in order to find properties of its stationary points, such as extrema or saddle points. In general, these properties take the form of differential equations that y has to satisfy in order to be a candidate extremum. For problems involving mechanical equilibria, variational calculus forms the link between Lagrangian mechanics and Newtonian mechanics, by letting us derive differential equations that a configuration has to satisfy in order to render a given energy potential stationary.

Apart from a general increase in simplicity, an advantage of working with Lagrangian mechanics is that it makes it easier to distinguish between stable and unstable equilibria, because they correspond to minima and saddle points of the energy potential, respectively. In Chapter 4, we will investigate a variational problem that is a version of the so-called "simplest problem" of the calculus of variations, with added isoperimetric constraints, to be introduced below. Finally, we review a set of sufficient conditions for a solution to be stable, based on the second variation.

An excellent introduction to this subject is given by Kot [Kot14]. The isoperimetric theory is discussed by Bolza [Bol02, Bol04], and its extension to multiple constraints by Manning [MRM98]. For an extensive modern introduction, see Hestenes [Hes66]. It bears mentioning that, as noted by Manning [MRM98], classical literature on the sufficiency of the strengthened Legendre and isoperimetric Jacobi conditions for more than one constraint is very sparse. Indeed, the only classical proof [Bol02] does not directly generalize to the case of multiple constraints.

3.3.1 The Simplest Problem

The "simplest problem" is to determine a function $y : (0, \ell) \to \mathbb{R}$ with $y(0) = y_0$ and $y(\ell) = y_\ell$ such that

$$F(y) = \int_0^\ell f(t, y(t), y'(t)) \, \mathrm{d}t$$

is locally minimal, for a given function f of sufficient regularity. Similar to finite-dimensional optimization problems, we say that y locally minimizes F if there exists $\varepsilon > 0$ such that for all $\tilde{y} : (0, \ell) \to \mathbb{R}$ with $\tilde{y}(0) = y_0$ and $\tilde{y}(\ell) = y_\ell$ and $\|\tilde{y} - y\| < \varepsilon$, we have $F(y) \leq F(\tilde{y})$.

An important difference to optimization problems over finite-dimensional normed vector spaces is that this definition depends on the choice of $\|\cdot\|$. For problems similar to the "simplest" one, the most common choices are the uniform norm $\|y\|_\infty = \sup |y|$ and the $\mathcal{C}^1$-norm $\|y\|_w = \sup |y| + \sup |y'|$, which are referred to as the *strong* and *weak* norm, respectively, in the context of variational calculus. Likewise, one calls y a *weak* or *strong* minimizer. Strong minimizers are always weak minimizers, but the converse is not the case, as one can demonstrate even on examples of "simplest" type, in which f is a polynomial.

This is not an issue for necessary conditions, such as the Euler–Lagrange equation derived below, because necessary conditions for a weak minimizer are also necessary conditions for a strong minimizer. Sufficient conditions, however, do not carry over automatically. We will discuss sufficient conditions for weak minimizers in Section 3.3.3.

The Euler–Lagrange Equation. To derive a necessary condition that a local minimizer has to satisfy, assume that $y : (0, \ell) \to \mathbb{R}$ was such a minimizer. Then, for arbitrary $\eta : (0, \ell) \to \mathbb{R}$ with $\eta(0) = 0 = \eta(\ell)$ and $\|\eta\|_w < \infty$, there is $\varepsilon_0 > 0$ small enough such that $F(y) \leq F(y + \varepsilon\eta)$ for all $\varepsilon \in (-\varepsilon_0, \varepsilon_0)$. In other words, the function $\varepsilon \mapsto F(y + \varepsilon\eta)$ has a minimum at zero. If this function is differentiable, then a necessary condition for this to be the case is $\frac{\mathrm{d}}{\mathrm{d}\varepsilon} F(y + \varepsilon\eta)|_{\varepsilon=0} = 0$. For F to have a local minimum at y, this needs to be satisfied for every η. Evaluating this expression for F as defined above gives

$$\begin{aligned} 0 = \frac{\mathrm{d}}{\mathrm{d}\varepsilon} F(y + \varepsilon\eta)\bigg|_{\varepsilon=0} &= \frac{\mathrm{d}}{\mathrm{d}\varepsilon} \int_0^\ell f(t, y(t) + \varepsilon\eta(t), y'(t) + \varepsilon\eta'(t))\,\mathrm{d}t\bigg|_{\varepsilon=0} \\ &= \int_0^\ell \left(\frac{\partial f}{\partial y}(t, y(t), y'(t)) \cdot \eta(t) + \frac{\partial f}{\partial y'}(t, y(t), y'(t)) \cdot \eta'(t) \right) \mathrm{d}t \\ &= \int_0^\ell \left(\frac{\partial f}{\partial y}(t, y(t), y'(t)) - \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial f}{\partial y'}(t, y(t), y'(t)) \right) \eta(t)\,\mathrm{d}t. \end{aligned}$$

The boundary term from using integration by parts in the last equality vanishes because $\eta(0) = 0 = \eta(\ell)$. Setting $U := \frac{\partial f}{\partial y} - \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial f}{\partial y'}$, we arrive at the necessary condition $0 = \int_0^\ell U(t, y(t), y'(t)) \cdot \eta(t)\,\mathrm{d}t$ for all η satisfying $\eta(0) = 0 = \eta(\ell)$, referred to as the *weak form* of the variational problem. The fundamental lemma of the calculus of variations says that this implies $U \equiv 0$, or

$$\frac{\partial f}{\partial y}(t, y(t), y'(t)) - \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial f}{\partial y'}(t, y(t), y'(t)) = 0$$

for all $t \in (0, \ell)$, known as the Euler–Lagrange equation, or *strong form* of the problem. If y satisfies the Euler–Lagrange equation, it is called a *stationary point* of F, regardless of whether it is a minimizer.

An immediate simplification can be made if f does not depend on $y(t)$, but only on t and $y'(t)$, i.e. $f = f(t, y'(t))$. In this case, the first term in the Euler–Lagrange equation vanishes, and we are left with $\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial f}{\partial y'} = 0$, which is equivalent to $\frac{\partial f}{\partial y'} = c$ for some $c \in \mathbb{R}$. Then, the necessary condition for y to be a local minimum is that there exists $c \in \mathbb{R}$ such that

$$\frac{\partial f}{\partial y'}(t, y'(t)) = c,$$

for all $t \in (0, \ell)$.

3.3.2 Isoperimetric Constraints

Isoperimetric constraints, also called integral constraints, are constraints on y of the form

$$G(y) = \int_0^\ell g(t, y(t))\,\mathrm{d}t = k$$

for some fixed $k \in \mathbb{R}$ and a given function g. The *constrained variational problem* is to find a function y with $y(0) = y_0$ and $y(\ell) = y_\ell$ that minimizes F locally among all functions that satisfy $G(y) = k$. As with the simplest problem, we will try to reduce this problem to a finite-dimensional one, so we can apply knowledge about constrained minimizers of problems on finite-dimensional spaces.

To derive a necessary condition, assume that y is a constrained minimizer, and that y is not a stationary point of G—this is a technical assumption which is required for the necessary

condition to hold. It follows that $T = \frac{\partial g}{\partial y}$ does not identically vanish. Choose η_1 and η_2 arbitrarily, but such that $\eta_i(0) = 0 = \eta_i(\ell)$ for $i = 1, 2$, and $\int_0^\ell T \cdot \eta_2 \neq 0$. This is possible because T does not identically vanish. Then, define

$$\tilde{y}(\varepsilon_1, \varepsilon_2) = y + \varepsilon_1\eta_1 + \varepsilon_2\eta_2, \quad \tilde{F}(\varepsilon_1, \varepsilon_2) = F(\tilde{y}(\varepsilon_1, \varepsilon_2)), \quad \text{and} \quad \tilde{G}(\varepsilon_1, \varepsilon_2) = G(\tilde{y}(\varepsilon_1, \varepsilon_2)).$$

By the same argument as for the simplest problem, $\tilde{F}$ has a constrained minimum at $(0, 0)$ subject to $\tilde{G} = k$. A computation similar to the derivation of the Euler–Lagrange equation shows that

$$\left.\frac{\partial \tilde{G}}{\partial \varepsilon_i}\right|_{\varepsilon_1, \varepsilon_2 = 0} = \int_0^\ell \frac{\partial g}{\partial y}\eta_i = \int_0^\ell T \cdot \eta_i,$$

so $\partial\tilde{G}/\partial\varepsilon_2 \neq 0$, and therefore $\nabla\tilde{G} \neq 0$. Under this condition, the theorem of Lagrange multipliers states that there exists $\lambda \in \mathbb{R}$ such that $(\varepsilon_1, \varepsilon_2) \mapsto \tilde{F}(\varepsilon_1, \varepsilon_2) + \lambda\tilde{G}(\varepsilon_1, \varepsilon_2)$ has a stationary point at $(0, 0)$. The first component of this condition yields

$$0 = \int_0^\ell \left(\frac{\partial(f + \lambda g)}{\partial y} - \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial(f + \lambda g)}{\partial y'}\right)\eta_1,$$

for arbitrary $\eta_1 : (0, \ell) \to \mathbb{R}$ with $\eta_1(0) = 0 = \eta_1(\ell)$.

This shows that a necessary condition for F to have a constrained minimum at y subject to $G(y) = k$ is that there exists $\lambda \in \mathbb{R}$ such that, for all $t \in (0, \ell)$,

$$\frac{\partial(f + \lambda g)}{\partial y}(t, y(t), y'(t)) - \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial(f + \lambda g)}{\partial y'}(t, y(t), y'(t)) = 0.$$

To handle $G(y) \in \mathbb{R}^d$, we can introduce a Lagrange multiplier $\lambda \in \mathbb{R}^d$, and substitute λg in the constrained Euler–Lagrange equation with $\langle \lambda, g \rangle$.

3.3.3 Stability of Solutions

So far, we have extracted information about candidate minimizers of a functional F using $\frac{\mathrm{d}}{\mathrm{d}\varepsilon}(F + \varepsilon\eta)|_{\varepsilon=0}$, which is called the *first variation* of F, in analogy to the first directional derivative in multi-variable calculus. The necessary condition described by the Euler–Lagrange equation is all the first variation has to give, and we have to rely on higher derivatives in order to distinguish actual local minimizers from other stationary points.

In finite-dimensional optimization problems, looking at the second derivatives of a function $f : \mathbb{R}^n \to \mathbb{R}$ often yields enough information to let us pin down local minima. If $f'(x)$ vanishes, it suffices to check whether all directional second derivatives at x are positive, i.e., $\frac{\mathrm{d}^2}{\mathrm{d}\varepsilon^2} f(x + \varepsilon d)|_{\varepsilon=0} > 0$ for all $d \neq 0$. Because the directional second derivatives in direction d can be computed from the Hessian H_f of f via $\langle d, H_f(x) \cdot d \rangle$, an equivalent criterion is that H_f be positive-definite at x. This reduces the problem of checking positivity for all directions to checking a finite number of inequalities. On the other hand, if H_f is not even positive semi-definite at x, this shows that x cannot be a local minimizer. This leaves only the case in which H_f is positive semi-definite but not positive definite. Then the second derivative does not contain enough information to determine the type of a stationary point.

In broad strokes, second-order necessary and sufficient conditions follow the same rules in variational problems. For problems of "simplest" type, a rigorous proof of the necessary and sufficient conditions can be given in fairly concise terms, and we will describe them here. The

proofs do not carry over directly to variational problems with isoperimetric constraints, but the proof of the necessary condition is short, so we will present it as well. For the proof of the sufficient condition, there is a somewhat lengthy classical proof for the case of one constraint [Bol02], but proofs for multiple constraints rely on modern theory which we do not cover here [Hes66].

There are two stability conditions, named after Legendre and Jacobi, that play a role in proving or disproving that a solution to the Euler–Lagrange equations is a minimizer to a variational problem. Each condition has a necessary and a sufficient version, loosely resembling the semi-positive definiteness and strict positive definiteness of the Hessian in the finite-dimensional case. The Legendre condition is the same for "simplest" and isoperimetric problems, but the Jacobi condition is more complex for the latter. The following table provides an overview of the conditions, whether they are necessary or sufficient, and whether we give a proof in the remainder of this section. We also cite references to self-contained proofs if available.

Name	Abbr	Condition	Type	Proof
Legendre condition	L	$R \geq 0$	necessary	yes; see [Kot14]
Strengthened Legendre condition	L^+	$R > 0$	-	-
Jacobi condition	J	$\sigma \geq \ell$	necessary	yes; see [Kot14, Mie15]
Strengthened Jacobi condition	J^+	$\sigma > \ell$	sufficient with L^+	yes; see [Kot14]
Isoperimetric Jacobi condition	J_{iso}	$\sigma_{\text{iso}} \geq \ell$	necessary	yes; [a] for 1 constraint see [Kne02, Bol03]
Strengthened isoperimetric Jacobi condition	$\mathsf{J}^+_{\text{iso}}$	$\sigma_{\text{iso}} > \ell$	sufficient with L^+	no; for 1 constraint see [Bol02]

[a] There is no classical published proof for the necessary isoperimetric Jacobi condition for more than one constraint. The proof presented here is original work, but requires one additional technical assumption, which is not generally required. This is similar to the first proof of the condition for one constraint by Kneser [Kne02], which left a special case open to be solved by Bolza [Bol03].

The Second Variation. Going back to the simplest problem, we want to check if a stationary point y is a weak local minimizer of F. A sufficient condition is that $\varepsilon \mapsto F(y+\varepsilon\eta)$ is minimal at $\varepsilon = 0$ for all choices of η with $\eta(0) = 0 = \eta(\ell)$. Expanding F as a Taylor series in ε yields

$$F(y+\varepsilon\eta) - F(y) = \varepsilon^2 \cdot \frac{1}{2} \frac{\mathrm{d}^2}{\mathrm{d}\varepsilon^2} F(y+\varepsilon\eta)\bigg|_{\varepsilon=0} + \mathcal{O}(\varepsilon^3) > 0,$$

because the linear term vanishes at y. Thus, an equivalent condition for a minimum at $\varepsilon = 0$ is that $\frac{\mathrm{d}^2}{\mathrm{d}\varepsilon^2} F(y+\varepsilon\eta)|_{\varepsilon=0} > 0$. A short computation reveals

$$\begin{aligned} \frac{\mathrm{d}^2}{\mathrm{d}\varepsilon^2} F(y+\varepsilon\eta)\bigg|_{\varepsilon=0} &= \int_0^\ell \left(P\eta^2 + 2Q\eta\eta' + R\eta'^2 \right), \\ \text{with} \quad P = \frac{\partial^2 f}{\partial y^2}, &\quad Q = \frac{\partial^2 f}{\partial y \partial y'}, \quad R = \frac{\partial^2 f}{\partial y'^2}. \end{aligned} \tag{3.4}$$

The Legendre Condition (L). A necessary condition which is sharper than the Euler–Lagrange equation alone is given by $R \geq 0$ on $(0, \ell)$. This was discovered first by Legendre in

1786 [Gol80]. We can show that the converse, namely $R(T) < 0$ for some $T \in (0, \ell)$, lets us construct η such that the second variation is negative.

To do this, we make a transformation to Eq. 3.4: Note that for any arbitrary differentiable function $w : (0, \ell) \to \mathbb{R}$, it holds that

$$\int_0^\ell (w'\eta^2 + 2w\eta\eta') = \int_0^\ell (w\eta^2)' = w\eta^2|_0^\ell = 0,$$

because η vanishes at 0 and ℓ. Therefore, we can rewrite

$$\frac{\mathrm{d}^2}{\mathrm{d}\varepsilon^2} F(y + \varepsilon\eta)\bigg|_{\varepsilon=0} = \int_0^\ell \left[(P + w')\eta^2 + 2(Q + w)\eta\eta' + R\eta'^2\right]. \tag{3.5}$$

From continuity of R (which follows from F twice continuously differentiable), there exists an interval (t_0, t_1) around T such that R is negative on all of (t_0, t_1). Choose as η a bump on this interval. This lets us shrink the integration interval to (t_0, t_1) and complete the square suggested by the last two terms in Eq. 3.5, which gives

$$\frac{\mathrm{d}^2}{\mathrm{d}\varepsilon^2} F(y + \varepsilon\eta)\bigg|_{\varepsilon=0} = \int_{t_0}^{t_1} \left\{ \left[(P + w') - \frac{(Q + w)^2}{R}\right] \eta^2 + R\left(\eta' + \frac{Q + w}{R}\eta\right)^2 \right\}.$$

If we choose w such that the term multiplying η^2 vanishes, we have almost completed the proof, as the remaining term is a product between a negative function and a square. Thus, we will attempt to solve

$$P + w' - \frac{(Q + w)^2}{R} = 0. \tag{3.6}$$

The Peano existence theorem guarantees that a solution w exists at least in a neighborhood around T. Should this neighborhood be smaller than (t_0, t_1), we can shrink (t_0, t_1) to be contained in the neighborhood, and adjust η accordingly. Finally, we arrive at

$$\frac{\mathrm{d}^2}{\mathrm{d}\varepsilon^2} F(y + \varepsilon\eta)\bigg|_{\varepsilon=0} = \int_{t_0}^{t_1} R\left(\eta' + \frac{Q + w}{R}\eta\right)^2,$$

which is non-positive, and only vanishes if $\eta' + \frac{Q+w}{R}\eta$ vanishes identically. However, this may only happen if η itself vanishes identically on (t_0, t_1), which is excluded by construction. This way, we have shown that $R(T) < 0$ contradicts the assumption that y is a minimizer, so $R \geq 0$ is a necessary condition.

One may be tempted to surmise that $R > 0$ is sufficient to claim a minimum. Then one would be in Legendre's good company, but mistaken nevertheless. It was only in 1836—50 years later—that Jacobi showed that $R > 0$ is only one of two inequalities that together comprise a sufficient condition.

The Strengthened Jacobi Condition (J$^+$). To arrive at a condition that lets us switch out "$\geq$" for "$>$" to turn a necessary into a sufficient condition—similar to the Hessian condition for finite-dimensional optimization problems—we have to append Legendre's condition with a second condition. The proof strategy for the necessity of the Legendre condition suggests that we would have had a sufficient condition if $R > 0$, and Eq. 3.6 had a solution w that was defined on all of $(0, \ell)$. Because then, we can make the transformation

$$\frac{\mathrm{d}^2}{\mathrm{d}\varepsilon^2} F(y + \varepsilon\eta)\bigg|_{\varepsilon=0} = \int_0^\ell R\left(\eta' + \frac{Q + w}{R}\eta\right)^2,$$

for arbitrary η, and see that the integral is strictly positive whenever the square term does not vanish identically. But this may only happen if $\eta \equiv 0$, because this is the unique solution of $\eta' + \frac{Q+w}{R}\eta = 0$ with the boundary conditions $\eta(0) = 0 = \eta(\ell)$.

A solution to Eq. 3.6, called a *Ricatti equation*, can be found by transforming the solution to a simpler linear equation, called the *Jacobi equation*,

$$\mathcal{S}u := (Ru')' + (Q' - P)u = 0. \tag{3.7}$$

Should this equation have a solution u with $u(t) \neq 0$ for all $t \in [0, \ell]$, we define

$$w(t) = -Q(t) - R(t)\frac{u'(t)}{u(t)},$$

and verify[3] that it solves Eq. 3.6. It remains to check whether Eq. 3.7 has a solution that never changes sign. A sufficient condition can be formulated in terms of the particular solution with boundary conditions $u(0) = 0$ and $u'(0) = 1$. If this solution has no zeros on $(0, \ell]$, then we can perturb the initial condition $u(0) = \varepsilon > 0$ without introducing additional zeros on this interval, because solutions to the Jacobi equation are continuous in the initial conditions. This yields a solution with no zeros on $[0, \ell]$.

To summarize: A solution y of the Euler–Lagrange equation is a local minimizer of F if $R > 0$ on all of $(0, \ell)$, and if the particular solution to $\mathcal{S}u = 0$ satisfying $u(0) = 0$ and $u'(0) = 1$ does not cross zero on $(0, \ell]$.

A point $\sigma \in (0, \ell]$ such that $u(\sigma) = 0$ is called a *conjugate point*. Thus, the strengthened Jacobi condition is often written as $\sigma > \ell$, to indicate that there is no conjugate point in the problem domain.

The Jacobi Condition (J). By weakening this condition to $\sigma \geq \ell$, we arrive at a necessary condition for a minimizer. Equivalently, if a conjugate point satisfies $\sigma < \ell$, then it can be shown that y is not a local minimizer. The case $\sigma = \ell$ is the one in which the second variation alone does not contain enough information to distinguish between different types of stationary points.

The proof of the necessity of this condition proceeds in two steps. In the first step, we show that $\sigma \leq \ell$ implies that there is a direction η in which the second variation $\frac{\mathrm{d}^2}{\mathrm{d}\varepsilon^2} f(y + \varepsilon\eta)|_{\varepsilon=0}$ vanishes. This tells us that y is in the very least not an unconditionally stable minimum, and has at most *marginal* stability—analogous to a Hessian which is at most positive semi-definite but not positive definite in a finite-dimensional optimization problem. In the second step, we show that $\sigma < \ell$ lets us perturb this direction to even make the second variation negative, which shows that y is not a minimizer.

For the first step, we go back to the original form of the second variation, Eq. 3.5, and show that it is closely connected to Eq. 3.7. This step uses the identity $\int_0^\ell 2Q\eta\eta' = \int_0^\ell -Q'\eta^2$, which follows from applying integration by parts,

$$\int_0^\ell 2Q\eta\eta' = \int_0^\ell \left(-2Q'\eta^2 - 2Q\eta\eta'\right),$$

[3] First, compute $w' = -Q' - \frac{1}{u}(Ru')' + R\frac{u'^2}{u^2}$, and $(Q + w)^2 = R^2\frac{u'^2}{u^2}$. Then, we see that

$$P + w' - \frac{(Q+w)^2}{R} = P - Q' - \frac{1}{u}(Ru')' = -\frac{1}{u}[(Q' - P)u + (Ru')'] = -\frac{1}{u}\mathcal{S}u = 0.$$

and moving the $Q\eta\eta'$ terms to the same side. Using the shorthand $W := P - Q'$, this lets us transform

$$\begin{aligned}\frac{\mathrm{d}^2}{\mathrm{d}\varepsilon^2}F(y+\varepsilon\eta)\bigg|_{\varepsilon=0} &= \int_0^\ell \left(P\eta^2 + 2Q\eta\eta' + R\eta'^2\right) = \int_0^\ell \left(W\eta^2 + R\eta'^2\right) \\ &= \int_0^\ell \left(W\eta^2 - (R\eta')'\eta\right) = -\int_0^\ell \eta\left((R\eta')' - W\eta\right) = -\int_0^\ell \eta\,\mathcal{S}\eta,\end{aligned} \tag{3.8}$$

which uses integration by parts again, between the first and second line.

To complete this step, we define η in such a way that the integrand $\eta\,\mathcal{S}\eta$ vanishes identically, causing the second variation to vanish. We do this by setting $\eta(t) = u(t)$ for $t \in [0, \sigma]$, and $\eta(t) = 0$ for $t \in (\sigma, \ell]$. Note that η satisfies $\eta(0) = 0 = \eta(\ell)$ and is continuous because $u(0) = 0 = u(\sigma)$. The integrand $\eta\,\mathcal{S}\eta$ vanishes identically because $\mathcal{S}\eta = \mathcal{S}u$ vanishes on $[0, \sigma]$, and η vanishes on $[\sigma, \ell]$.

In the second step, we perturb this choice of η by adding a positive smooth function that vanishes at the endpoints, for example $h(t) = t(\ell - t)$, multiplied by a small positive factor $\iota > 0$. Then, we show that the second variation is negative in direction $\eta + \iota h$ for ι small enough [Mie15, Thm. 3.2.8]. This step is more tedious than the first and is mostly included for completion's sake. No important insights will be missed upon skipping ahead to *"Isoperimetric Constraints"*.

The proof consists mostly of a series of manipulations and applications of integration by parts to the quantity $\frac{\mathrm{d}^2}{\mathrm{d}\varepsilon^2}F(y + \varepsilon(\eta + \iota h))|_{\varepsilon=0}$, to show that it is negative for ι small enough. Two of the steps rely on results that we discuss separately. The first is the identity $-\int_0^\sigma u\,\mathcal{S}h = R(\sigma)u'(\sigma)h(\sigma)$, which can be shown using elementary calculus.[4] The second is the strict inequality $u'(\sigma) < 0$: Because σ is the first root of u, and u is positive on $(0, \sigma)$, we must have $u'(\sigma) \leq 0$. But if $u'(\sigma) = 0$, it follows from $u(\sigma) = 0$ that $u \equiv 0$ solves Eq. 3.7. Since this equation has a unique solution, this is the only solution, which contradicts the initial condition $u'(0) = 1$. Therefore we have $u'(\sigma) < 0$.

Now we are ready to show that the second variation is negative in direction $\xi := \eta + \iota h$ for ι small enough. Written in full, this direction is defined by

$$\xi(t) = \begin{cases} u(t) + \iota h(t) & \text{if } t \in (0, \sigma], \\ \iota h(t) & \text{otherwise.} \end{cases}$$

We see that ξ is continuous because $u(\sigma) = 0$ but not differentiable at σ. However, it has a left derivative $\xi'_-(\sigma) = u'(\sigma) + \iota h'(\sigma)$ and a right derivative $\xi'_+(\sigma) = \iota h'(\sigma)$. To start with, we split the integral defining the second variation in direction ξ into contributions from $(0, \sigma)$ and (σ, ℓ). Then, we apply the same transformation as in Eq. 3.8 to both integrals, but this

[4] To show the identity, compute

$$\begin{aligned}-\int_0^\sigma u\,\mathcal{S}h &= \int_0^\sigma (Wuh - u(Rh')') \overset{(1)}{=} \int_0^\sigma (Wuh + Ru'h') = \int_0^\sigma (Wuh + (Ru'h)' - (Ru')'h) \\ &= \int_0^\sigma ((Ru'h)' - h\,\mathcal{S}u) \overset{(2)}{=} Ru'h\big|_0^\sigma \overset{(3)}{=} (Ru'h)(\sigma).\end{aligned}$$

(1) uses integration by parts, and $u(0) = 0 = u(\sigma)$. (2) uses that $\mathcal{S}u$ vanishes identically and the fundamental theorem of calculus. (3) uses that $h(0) = 0$.

time, boundary terms remain because $R\xi'\xi$ does not vanish at σ:

$$\begin{aligned}\frac{\mathrm{d}^2}{\mathrm{d}\varepsilon^2}F(y+\varepsilon\xi)\bigg|_{\varepsilon=0} &= \int_0^\sigma \left(W\xi^2 + R\xi'^2\right) + \int_\sigma^\ell \left(W\xi^2 + R\xi'^2\right) \\ &= -\int_0^\sigma \xi\,\mathcal{S}\xi + (R\xi'_-\xi)(\sigma) - \int_\sigma^\ell \xi\,\mathcal{S}\xi - (R\xi'_+\xi)(\sigma) = (*)\end{aligned} \tag{3.9}$$

Next, we insert the definition of ξ, cancel terms, and consolidate by using linearity of $\mathcal{S}$:

$$\begin{aligned}(*) &= -\int_0^\sigma (u+\iota h)\mathcal{S}(u+\iota h) + [R(u'+\iota h')\iota h](\sigma) - \iota^2\int_\sigma^\ell h\,\mathcal{S}h - \iota^2(Rh'h)(\sigma) \\ &= -\iota\int_0^\sigma (u+\iota h)\mathcal{S}h + \iota(Ru'h)(\sigma) - \iota^2\int_\sigma^\ell h\,\mathcal{S}h \\ &= \iota\left((Ru'h)(\sigma) - \int_0^\sigma u\,\mathcal{S}h - \iota\int_0^\ell h\,\mathcal{S}h\right) = (*),\end{aligned} \tag{3.10}$$

which, using our identity for $\int_0^\sigma u\,\mathcal{S}h$, becomes

$$(*) = \iota\left(2(Ru'h)(\sigma) - \iota\int_0^\ell h\,\mathcal{S}h\right).$$

Because $R, h > 0$ and $u'(\sigma) < 0$, the term $2(Ru'h)(\sigma)$ is strictly negative. By choosing ι small enough, it can be ensured that this term dominates and the entire expression is negative.

Isoperimetric Constraints. The first step in establishing second-order conditions on minimizers subject to isoperimetric constraints is to find the analogue of the unconstrained condition that the second variation be non-negative. Similar to the way we derived the constrained Euler–Lagrange equation, this can be done by reducing the problem to a finite-dimensional one by choosing η and introducing a parameter ε. The second-order necessary conditions for a constrained minimizer $x \in \mathbb{R}^n$ of $f : \mathbb{R}^n \to \mathbb{R}$ subject to $g(x) = 0$ with $g : \mathbb{R}^n \to \mathbb{R}^d$ are given by

$$\langle d, H_{f+\langle\lambda,g\rangle}(x)\cdot d\rangle \geq 0 \quad \text{for all} \quad d \in \mathbb{R}^n \text{ with } J_g \cdot d = 0,$$

where λ is the Lagrange multiplier associated with the stationary point x, and J_g is the Jacobian matrix of g. In other words, the second directional derivative of $f + \lambda g$ has to be non-negative in all directions orthogonal to the constraint tangents. A sufficient condition is obtained by replacing "$\geq$" with "$>$".

For isoperimetric variational problems, this condition can be translated directly from the finite-dimensional one, by replacing the second directional derivative with the second variation, and J_g with $\partial g/\partial y$,

$$\frac{\mathrm{d}^2}{\mathrm{d}\varepsilon^2}F^*(y+\varepsilon\eta)\bigg|_{\varepsilon=0} \geq 0 \quad \text{for all} \quad \eta : (0,\ell) \to \mathbb{R} \text{ with } \eta(0) = 0 = \eta(\ell) \text{ and } \int_0^\ell \frac{\partial g}{\partial y}\eta = 0,$$

where we have abbreviated $F^* = F + \langle\lambda, G\rangle$. Similar to the unconstrained problem, we define P, Q, and R by replacing f with $f^* = f + \langle\lambda, g\rangle$, so we reproduce the formula

$$\frac{\mathrm{d}^2}{\mathrm{d}\varepsilon^2}F^*(y+\varepsilon\eta)\bigg|_{\varepsilon=0} = \int_0^\ell \left(P\eta^2 + 2Q\eta\eta' + R\eta'^2\right) = -\int_0^\ell \eta\,\mathcal{S}\eta,$$
$$\text{with} \quad P = \frac{\partial^2 f^*}{\partial y^2}, \quad Q = \frac{\partial^2 f^*}{\partial y \partial y'}, \quad R = \frac{\partial^2 f^*}{\partial y'^2}.$$

The necessity proof of the Legendre condition $R = \frac{\partial^2 f^*}{\partial y'^2} \geq 0$ follows the unconstrained case line by line, except that η must be chosen on (t_0, t_1) to satisfy the constraint tangent orthogonality condition $\int_{t_0}^{t_1} \frac{\partial g}{\partial y} \eta = 0$.

The isoperimetric Jacobi condition is more complicated than the one in the unconstrained setting, and we will introduce a few derived quantities before stating it in full. Assume that y is a constrained minimizer of F subject to $G = k$. Define the constraint tangents $T_i = \langle e_i, \frac{\partial g}{\partial y} \rangle : (0, \ell) \to \mathbb{R}$ for $i = 1, \dots, d$, and let

$$\begin{aligned} \mathcal{S}u = 0 \quad &\text{with} \quad u(0) = 0, u'(0) = 1, \\ \mathcal{S}v_i = T_i \quad &\text{with} \quad v_i(0) = 0, v_i'(0) = 1. \end{aligned}$$

the solutions to the homogeneous and inhomogeneous Jacobi equations with the constraint tangents as right-hand sides. Furthermore, set $M_i(t) = \int_0^t T_i u$ and $N_{ij}(t) = \int_0^t T_i v_j$ for all $i, j = 1, \dots, d$. Define the *constrained stability matrix* $Z : (0, \ell) \to \mathbb{R}^{(d+1)\times(d+1)}$ by

$$Z = \begin{pmatrix} u & v_1 & \cdots & v_d \\ M_1 & N_{11} & \cdots & N_{1d} \\ \vdots & \vdots & \ddots & \vdots \\ M_d & N_{d1} & \cdots & N_{dd} \end{pmatrix} = \begin{pmatrix} u & v^t \\ M & N \end{pmatrix}.$$

A point σ_{iso} such that $\det Z(\sigma_{\text{iso}}) = 0$ is called an *isoperimetric conjugate point*.

The Isoperimetric Jacobi Condition ($\mathbf{J_{iso}}$). A necessary condition for y to be a constrained minimum is $\sigma_{\text{iso}} \geq \ell$, i.e., there be no isoperimetric conjugate point in the open interval $(0, \ell)$. There is also a technical assumption required for an elementary proof to work, which we mention below. Like in the unconstrained setting, the proof proceeds in two steps: The first establishes a direction in which the second variation vanishes, and the second makes a perturbation that causes it to become strictly negative.

For the first step, we define the direction η by the following construction: Let $\bar{Z}(t) = \begin{pmatrix} u(t) & v(t)^t \\ M(\sigma_{\text{iso}}) & N(\sigma_{\text{iso}}) \end{pmatrix}$, and define $\eta(t) = \det \bar{Z}(t)$ for $t \in [0, \sigma]$ and $\eta(t) = 0$ otherwise. Because $\eta(\sigma_{\text{iso}}) = \det \bar{Z}(\sigma_{\text{iso}}) = \det Z(\sigma_{\text{iso}}) = 0$, we have continuity of η. We need to show that η is an admissible variation, in the sense that it satisfies $\int_0^l T_k \eta = 0$ for all $k = 1, \dots, d$. To see this, compute

$$\begin{aligned} \int_0^\ell T_k \eta &= \int_0^{\sigma_{\text{iso}}} \det \begin{pmatrix} T_k(t)u(t) & T_k(t)v(t)^t \\ M(\sigma_{\text{iso}}) & N(\sigma_{\text{iso}}) \end{pmatrix} \mathrm{d}t \\ &= \det \begin{pmatrix} M_k(\sigma_{\text{iso}}) & N_{k1}(\sigma_{\text{iso}}) \cdots N_{kd}(\sigma_{\text{iso}}) \\ M(\sigma_{\text{iso}}) & N(\sigma_{\text{iso}}) \end{pmatrix} = 0, \end{aligned}$$

where the last equality follows from the fact that the first and $(1+k)$-th row of the matrix are equal. The next step is to represent $\mathcal{S}\eta$ as a linear combination of the constraint tangents,

$$\mathcal{S}\eta = \det \begin{pmatrix} \mathcal{S}u & \mathcal{S}v_1 \cdots \mathcal{S}v_d \\ M(\sigma_{\text{iso}}) & N(\sigma_{\text{iso}}) \end{pmatrix} = \det \begin{pmatrix} 0 & T_1 \cdots T_d \\ M(\sigma_{\text{iso}}) & N(\sigma_{\text{iso}}) \end{pmatrix} = \sum_{i=1}^d \mu_i T_i,$$

for suitable $\mu_1, \dots, \mu_d \in \mathbb{R}$. Then it immediately follows that

$$\frac{\mathrm{d}^2}{\mathrm{d}\varepsilon^2} F^*(y + \varepsilon\eta)\Big|_{\varepsilon=0} = -\int_0^\ell \eta\, \mathcal{S}\eta = -\int_0^\ell \eta \left(\mathcal{S}\eta - \sum_{i=1}^d \mu_i T_i \right) = 0.$$

This shows that y is not an unconditionally stable constrained minimum.

The second step proceeds the same way as in the unconstrained setting, with a few more calculations in order to eliminate terms arising from $\det \bar{Z}(\sigma_{\text{iso}})$. As before, we perturb η in order to produce a new direction $\xi(t) = \eta(t) + \iota h(t)$, with $h(0) = 0 = h(\ell)$. To maintain the property $\int_0^\ell T_k \xi = 0$ for all $k = 1, \dots, d$, the function h has to be chosen to satisfy $\int_0^\ell T_k h = 0$ as well. Some steps of the calculation require us to place further constraints on h, such as $h'(\sigma_{\text{iso}}) = 0$ and $\int_0^{\sigma_{\text{iso}}} T_k h = 0$. However, such a function h is still guaranteed to exist because it is chosen from an infinite-dimensional function space arising as the orthogonal complement to the finite-dimensional subspace defined by these constraints.

The same transformations as in Eqs. 3.9 and 3.10 lead to

$$\begin{aligned}\frac{\mathrm{d}^2}{\mathrm{d}\varepsilon^2} F^*(y + \varepsilon\xi)\bigg|_{\varepsilon=0} &= -\int_0^\ell \xi\, \mathcal{S}\xi + \iota(R\eta'_- h)(\sigma) \\ &= -\iota^2 \int_0^\ell h\, \mathcal{S}h - \iota \int_0^\sigma (h\, \mathcal{S}\eta + \eta\, \mathcal{S}h) + \iota(R\eta'_- h)(\sigma),\end{aligned}$$

where we have omitted the subscript "iso". The middle term can be transformed as follows,

$$\begin{aligned}-\int_0^\sigma (h\, \mathcal{S}\eta + \eta\, \mathcal{S}h) &= -\int_0^\sigma \left[h \det \begin{pmatrix} 0 & T^t \\ M(\sigma) & N(\sigma) \end{pmatrix} + \det \begin{pmatrix} u & v^t \\ M(\sigma) & N(\sigma) \end{pmatrix} \mathcal{S}h \right] \\ &= \det \begin{pmatrix} -\int_0^\sigma u\, \mathcal{S}h & -\int_0^\sigma (hT^t + \mathcal{S}h\, v^t) \\ M(\sigma) & N(\sigma) \end{pmatrix} = (*)\end{aligned}$$

We know from footnote 4 that the top-left entry in this last matrix simplifies to $-\int_0^\sigma u\, \mathcal{S}h = (Ru'h)(\sigma)$. The expressions $\int_0^\sigma T_i h$ vanish by construction of h. This leaves $-\int_0^\sigma v_i\, \mathcal{S}h$, which can be transformed in exactly the same way as in footnote 4 to yield $(Rv_i' h)(\sigma)$. This gives

$$\begin{aligned}(*) &= \det \begin{pmatrix} (Ru'h)(\sigma) & (R(v')^t h)(\sigma) \\ M(\sigma) & N(\sigma) \end{pmatrix} \\ &= R(\sigma)h(\sigma) \det \begin{pmatrix} u'(\sigma) & (v')^t(\sigma) \\ M(\sigma) & N(\sigma) \end{pmatrix} = (R(\det \bar{Z})' h)(\sigma) = (R\eta'_- h)(\sigma).\end{aligned}$$

Thus, we find that the second variation can again be represented as

$$\frac{\mathrm{d}^2}{\mathrm{d}\varepsilon^2} F^*(y + \varepsilon\xi)\bigg|_{\varepsilon=0} = \iota \left(2(R\eta'_- h)(\sigma_{\text{iso}}) - \iota \int_0^\ell h\, \mathcal{S}h \right),$$

which gives the desired result under the assumption $\eta'_-(\sigma_{\text{iso}}) = (\det \bar{Z})'(\sigma_{\text{iso}}) \neq 0$.

The Strengthened Isoperimetric Jacobi Condition ($\mathbf{J}^+_{\text{iso}}$). Finally, a sufficient condition is given by strengthening the isoperimetric Jacobi condition to $\sigma_{\text{iso}} > \ell$ to exclude the coincidence of the first conjugate point with the right endpoint, together with the strengthened Legendre condition $R > 0$. As noted in the introduction, there is no published classical proof for this condition that generalizes to more than one isoperimetric constraint. This case is discussed at length by Manning [MRM98], who cites Bolza [Bol02] as the only classical source giving a proof for a single constraint. A proof with more than one constraint can be found in Hestenes [Hes66], but it uses a formalism that is too far removed from the one used in this thesis to be included here.

3.4 The Adjoint Method

Broadly speaking, the adjoint method (or adjoint state method) is a way of calculating the derivative of the objective function in constrained optimization problems, more efficiently than by repeated application of the chain rule [Bra19, CLPS03]. Typically, this method is applied to optimization problems constrained by differential equations, but the main idea can be illustrated via a standard nonlinear optimization problem with nonlinear constraints,

$$\min_{x\in\mathbb{R}^n, p\in\mathbb{R}^m} f(x) \text{ s.t. } g(p,x)=0, \tag{3.11}$$

with $f:\mathbb{R}^n\to\mathbb{R}$ and $g:\mathbb{R}^m\times\mathbb{R}^n\to\mathbb{R}^n$, supposing that $J_{g,x}(p,x)$ is invertible for all (p,x) such that $g(p,x)=0$. The constraints are of the form discussed in Section 3.2, so given (p_0,x_0) with $g(p_0,x_0)=0$, we are guaranteed the existence of a function $x(p)$ such that $x(p_0)=x_0$ and $g(p,x(p))=0$ for all (p,x) sufficiently close to (p_0,x_0). Furthermore, we can compute $J_x(p)=-\left(J_{g,x}(p,x)\right)^{-1}J_{g,p}(p,x)$.

We see that choosing $p_0\in\mathbb{R}^m$ uniquely defines $x_0\in\mathbb{R}^m$ (at least locally) such that $g(p_0,x_0)=0$. Therefore, a common way of solving Eq. 3.11 is to interpret p as the *control variable* and x as the dependent variable, and run an iterative gradient-based *unconstrained* optimization algorithm on the problem "$\min_{p\in\mathbb{R}^m}(f\circ x)(p)$". Using what we know, we can compute the gradient of $f\circ x$ with the chain rule from

$$J_{f\circ x}(p)=J_{f,x}(x(p))\cdot J_x(p)=-J_{f,x}(x(p))\left[J_{g,x}(p,x(p))^{-1}J_{g,p}(p,x(p))\right].$$

For many applications, m and n may be very large, for example if x and p are discretizations of functions on $\mathbb{R}^d$. Often, however, $J_{g,p}$ and $J_{g,x}$ will have a number of non-zeros which is linear in m or n, so they can be represented efficiently as sparse matrices. The same will usually not hold for J_x, which is a dense n-by-m matrix, so it would be best to avoid computing it. For this problem, it is easy to see that this can be achieved by rewriting the equation as

$$J_{f\circ x}(p)=-\left[J_{f,x}(x(p))\cdot J_{g,x}(p,x(p))^{-1}\right]J_{g,p}(p,x(p)),$$

so, by solving the linear system $J_{g,x}^t(p,x(p))\cdot\lambda=J_{f,x}^t(x(p))$ and then computing $J_{f\circ x}(p)=-\langle\lambda,J_{g,p}(p,x(p))\rangle$. This is surely more efficient, because we solve a linear system with a single right-hand side instead of m right-hand sides, and avoid storing the n-by-m solution.

The same idea can be applied to more complicated optimization problems and variational problems, with constraints given by PDEs, ODEs, or of isoperimetric type. The resulting *adjoint equations*—which, in the example above, is the linear system for computing λ—can then be solved to yield the gradient of the objective function with respect to the control parameters. Unlike the original equations, the adjoint equations are always linear, and may take the form of finite-dimensional linear systems or differential equations, depending on the original problem.

3.4.1 Céa's Method

The adjoint equations can be derived for a given optimization problem using Céa's method [Sha18, Cea86], a general strategy for doing the right kinds of substitutions and transformations in order to achieve the goal. It is easiest to first illustrate the method on a concrete example, before explaining how it works more generally.

Let us consider the constrained optimization problem

$$\min_{g,u:(0,1)\to\mathbb{R}} \int_0^1 u^2 \text{ s.t. } u'' = g, u(0) = 0, u'(1) = 1.$$

Since u is uniquely determined by g, we can view g as the control parameter, and consider the objective as a functional $F(g) = \int_0^1 u^2$. To perform gradient-based optimization, we want to identify $\delta F[g; \delta g]$. The goal of the adjoint method is to avoid computing intermediate quantities such as δu in the process.

Céa's method achieves this by rewriting F as

$$F = \int_0^1 u^2 + \int_0^1 \bar{u}(u'' - g),$$

in which we can choose the function $\bar{u}$ arbitrarily without changing F, because $u'' = g$. If we apply the δ-operator to F, this will result in the appearance of terms involving δu at first. But, as we will see shortly, $\bar{u}$ can be chosen to cancel out δu, which simplifies the computation of δF.

The variation of F is given by

$$\begin{aligned}
\delta F &= \int_0^1 (2u\,\delta u + \bar{u}(\delta u'' - \delta g)) \\
&= (\bar{u}\,\delta u' - \bar{u}'\delta u)|_0^1 + \int_0^1 ((2u + \bar{u}'')\,\delta u - \bar{u}\,\delta g) \\
&= -\bar{u}'(1)\,\delta u(1) - \bar{u}(0)\,\delta u'(0) + \int_0^1 ((2u + \bar{u}'')\,\delta u - \bar{u}\,\delta g)\,,
\end{aligned}$$

where he have first used integration by parts twice, and then the boundary conditions $\delta u(0) = 0$ and $\delta u'(1) = 0$. We can see that choosing $\bar{u}$ as the solution to the *adjoint equation*

$$\bar{u}'' = -2u, \quad \text{s.t.} \quad \bar{u}(0) = 0, \quad \bar{u}'(1) = 0,$$

simplifies $\delta F = -\int_0^1 \bar{u}\,\delta g$, and δu no longer appears. This means that we can compute δF by first solving the adjoint equation for $\bar{u}$, and then integrating it against δg. Equivalently, $-\bar{u}$ is the L^2-gradient of F, because $\delta F = \langle -\bar{u}, \delta g \rangle_{L^2}$.

The general recipe of Céa's method is to append to the objective function F a sum of inner products between quantities known to be zero (expressions involving differential operators, constraints, etc.) and adjoint variables, which we denote by overbars. For differential equations, such as $u'' = g$, adjoint variables take the form of functions, and the L^2-inner product is used. For integral constraints, adjoint variables are scalars, and we use the standard Euclidean inner product. Then, we manipulate δF in order to isolate the adjoint equations, which we can solve in order to simplify δF.

3.5 Numerical Optimization

At various points of this thesis, we will be confronted with minimization problems of the type

$$\min_{x\in\mathbb{R}^n} f(x) \text{ s.t. } x \in A \subset \mathbb{R}^n$$

with $f : D \to \mathbb{R}$ at least once, but usually twice, continuously differentiable, where $A \subset D \subset \mathbb{R}^n$. Often, we will have $A = \mathbb{R}^n = D$, or A large enough to cover all x we are likely

to explore, in which case the minimization problem is called *unconstrained*. At other times, A will be given implicitly by $A = \{x \in \mathbb{R}^n : a(x) = 0\}$ with $a : \mathbb{R}^n \to \mathbb{R}^d$ such that A is an $(n-d)$-dimensional submanifold of $\mathbb{R}^n$. In this case, it makes a practical difference for numerical optimization whether $D = A$, or D contains at least a neighborhood of A. These types of optimization problems are called *equality-constrained*.

Algorithms for numerical optimization can be categorized into those that only employ evaluations of f and J_f, called first-order, and those that also evaluate the Hessian H_f, called second-order. However, some strategies such as line-search algorithms can be adapted to either category. As a general rule, we use second-order algorithms for most small-scale problems with few variables, and for mid-scale problems if the Hessian is sparse and can be computed easily.

The fields of unconstrained and constrained gradient-based optimization are so vast that we do not attempt to give a complete overview here. Instead, we discuss only the methods that are used for the work presented in this thesis, and point out the most important convergence results if available [NW06]. We use most of these algorithms in one of their established variants, except for projected gradient descent, which we modify to suit our problem as described below.

3.5.1 Line-Search Methods

This collection of methods, primarily for unconstrained optimization, produces a sequence of iterates $x_0, x_1, \ldots$ by first determining, at every $k = 0, 1, \ldots$, a *search direction* $p_k \in \mathbb{R}^n$, then a *step size* $\alpha_k > 0$, and finally setting $x_{k+1} = x_k + \alpha_k p_k$. To guarantee the existence of a step size such that $f(x_{k+1}) < f(x_k)$, we require $J_f(x_k) \cdot p_k < 0$. The simplest such choice is $p_k = -J_f^t(x_k)$, the negative of the gradient, and gives rise to the algorithm of *steepest descent*.

To determine α_k, there is a wealth of options, the simplest one being the *backtracking line search* algorithm. Here, one chooses a problem-specific $\bar{\alpha} > 0$ that represents the maximal step size, a contraction factor $\rho \in (0, 1)$, and a *sufficient decrease* constant $c \in (0, 1)$, usually $c \approx 10^{-3}$. Then, compute for $i = 0, 1, \ldots$ the decrease in function value $\beta_k^i = f(x_k) - f(x_k + \rho^i \bar{\alpha} p_k)$, and pick $\alpha_k = \rho^i \bar{\alpha}$ where i is the smallest index such that $\beta_k^i > -c \cdot J_f(x_k) \cdot \rho^i \bar{\alpha} p_k$. This inequality guarantees that the decrease is at least within a factor of c of what can be expected based on the gradient at x_k. Backtracking line search always terminates after a finite number of steps, because this inequality is satisfied for all large enough i.

For line-search methods, convergence results can be obtained under very mild conditions that are independent from the way in which the descent direction and step size are computed. Assume only that the angle between the gradient and the descent direction is bounded away from a right angle by a constant $\delta > 0$ which is independent of k, i.e.,

$$\frac{-J_f(x_k) \cdot p_k}{\|J_f^t(x_k)\| \|p_k\|} > \delta,$$

and that the step size is small enough to represent a sufficient decrease in function value compared to what can be expected from gradient information, but large enough so the sequence of iterates does not get stuck. One way of formalizing this is

$$f(x_k) + (1-c) g_k \leq f(x_k + \alpha_k p_k) \leq f(x_k) + c g_k, \quad \text{with} \quad g_k = J_f(x_k) \cdot \alpha_k p_k,$$

for some $c \in (0, 1/2)$, called the *Goldstein conditions*. This alone suffices to conclude $\lim_{k \to \infty} \|J_f^t(x_k)\| = 0$, due to a result of Zoutendijk [Zou70]. There is no general guarantee

that x_k will converge to a local minimum, rather than a saddle point. However, only local minima are attracting for line-search methods, so convergence to a saddle point occurs with zero probability.

The rate of convergence, which is a good indication of how many steps will be needed to approach a local minimum to high precision, depends strongly on the choice of search direction. Under the same mild conditions as in Zoutendijk's result, we can only expect linear convergence,

$$f(x_{k+1}) - f(x^*) \leq r \cdot (f(x_k) - f(x^*)),$$

for a problem-dependent $r \in (0, 1)$, and for large enough k. Unfortunately, r is extremely sensitive to the conditioning and scaling of the problem, and tends to be very close to 1 for higher-dimensional problems.

3.5.2 Newton and Quasi-Newton Methods

Two ways of improving the convergence rate is to either use second-order information directly, or to approximate second-order information from first-order information at multiple iterates. Newton methods are based on choosing the iteration $p_k = -H_f^{-1}(x_k) \cdot J_f^t(x_k)$ with $\alpha_k = 1$ whenever possible. This is because the sequence x_k based on this choice of search direction and step size converges quadratically to a local minimum,

$$\|x_{k+1} - x^*\| \leq r \cdot \|x_k - x^*\|^2,$$

albeit under very strict conditions, among them that x_0 be close enough to x^*. Whether x_0 is indeed close enough to x^* is usually difficult to determine a priori, because it is related to the Lipschitz constant of H_f in a neighborhood of x^*.

A compromise is to use the Newton method embedded in a line-search method, but this comes with a new challenge: the search direction $p_k = -H_f^{-1}(x_k) \cdot J_f^t(x_k)$ is only guaranteed to be a descent direction if $H_f(x_k)$ is positive definite, which need not be the case far away from a local minimum. In this case, the search direction needs to be modified, and this is often done by first modifying the Hessian $\hat{H}_f(x_k) \approx H_f(x_k)$ to be positive definite, and using it in the computation of p_k instead of $H_f(x_k)$.

If the exact Hessian is not available or too expensive to compute, one often employs a quasi-Newton method. In this class of methods, an approximation B_k of the Hessian is maintained from iterate to iterate, and updated to reflect new information that is gathered from evaluations of the gradient. The most popular method, named BFGS for its discoverers [NW06, p. 136], updates B_k in such a way that the quadratic model

$$\hat{f}_{k,B_k}(x) = f(x_k) + J_f(x_k) \cdot (x - x_k) + \frac{1}{2}\langle x - x_k, B_k \cdot (x - x_k)\rangle$$

retains a good gradient approximation at the previous and the current iterate, i.e.,

$$J_{\hat{f}_{k,B_k}}(x_{k-1}) = J_f(x_{k-1}) \quad \text{and} \quad J_{\hat{f}_{k,B_k}}(x_k) = J_f(x_k).$$

The second condition is automatic, from the definition of $\hat{f}_k$, but the first needs to be enforced via the update from B_{k-1} to B_k. This alone does not make B_k unique, but uniqueness is enforced via the auxiliary minimization problem

$$B_k = \arg\min_B \|B^{-1} - B_{k-1}^{-1}\|_{\mathsf{F}} \text{ subject to } B = B^t \text{ and } J_{\hat{f}_{k,B}}(x_{k-1}) = J_f(x_{k-1}).$$

This way of defining B_k is popular because there is a closed-form solution for computing B_k^{-1} from B_{k-1}^{-1}. Thus it suffices to keep track of only these inverses, making the computation of $p_k = -B_k^{-1} \cdot J_f^t(x_k)$ very cheap. While quasi-Newton methods do not generally reach the quadratic convergence of Newton methods, they can be proven to converge super-linearly close to a local minimizer,

$$\|x_{k+1} - x^*\| \leq r \cdot \|x_k - x^*\|^{1+\alpha},$$

for some problem-dependent $\alpha > 0$, and k large enough.

3.5.3 Projected Gradient Descent

The standard projected gradient method attempts to solve the general constrained minimization problem

$$\min_{x \in \mathbb{R}^n} f(x) \text{ s.t. } x \in A \subset \mathbb{R}^n,$$

and is applicable even when $f : A \to \mathbb{R}$, i.e., when f cannot be evaluated outside of the feasible region. Every iteration in projected gradient descent consists of two steps to compute x_{k+1} from x_k,

$$y_k = x_k - \alpha_k \cdot J_f^t(x_k),$$
$$x_{k+1} = \arg\min_{x \in A} \|y_k - x\|,$$

a standard line-search step and a step that orthogonally projects the resulting point back onto A if necessary. Convergence results for projected gradient descent variants are less broad than those for unconstrained line-search methods, and rely on specific ways of computing the step size, on convexity of A, and on convexity (or at least quasi-convexity) of f [BL10].

If the feasible set is given implicitly as $A = \{x \in \mathbb{R}^n : a(x) = 0\}$ for some $a : \mathbb{R}^n \to \mathbb{R}^d$, and $y_k \notin A$, then x_{k+1} satisfies

$$a(x_{k+1}) = 0, \quad x_{k+1} - y_k = J_a(x_{k+1}) \cdot \lambda$$

for some $\lambda \in \mathbb{R}^d$. If a has no special structure, such as being linear, this is a system of $n + d$ non-linear equations, and can often be solved efficiently using Newton's method. This is because y_k is obtained by stepping away from $x_k \in A$, so at least for small step sizes, y_k will be close enough to A for Newton's method to succeed. Applying Newton's method here implies computing the Jacobian of the function $x_{k+1} \mapsto J_a(x_{k+1}) \cdot \lambda$, which will involve the Hessian of a.

In Section 4.6.2, we use a local–global optimization algorithm that relies on a version of projected gradient descent in its local step. To avoid computing the Hessian of a in the projection step, we make two changes to the algorithm. First, instead of using the search direction $-J_f^t(x_k)$ directly, we modify it by orthogonally projecting it onto the null space of $J_a(x_k)$, or equivalently, onto the tangent space of A at x_k. As a result, the shortest distance between A and y_k becomes quadratic in the step size instead of linear, generally keeping y_k closer to A. If x_k is close to a constrained minimizer, and thus $J_f^t(x_k)$ is almost orthogonal to the tangent space, then the search vector almost vanishes after this projection.

Second, instead of computing the exact orthogonal projection from y_k onto A, we compute an approximation via the following algorithm. Let $z_k^0 = y_k$ and compute z_k^{i+1} as the orthogonal projection of z_k^i onto the zero-level set of the first-order approximation of a at z_k^i,

$$z_k^{i+1} = \arg\min_x \|x - z_k^i\| \text{ s.t. } 0 = a(z_k^i) + J_a(z_k^i) \cdot (x - z_k^i),$$

which can be computed by solving a linear d-by-d system. The sequence $z_k^0, z_k^1, \ldots$ constitutes the application of the underdetermined variant of Newton's method to find a point on A, with y_k as a starting point. If a is linear, then this method reproduces the exact orthogonal projection of y_k onto A, and otherwise, an approximation of it, which proved good enough for the projected gradient descent algorithm to make progress in our problem.

CHAPTER 4

The Design Space of Plane Elastic Curves

4.1 Introduction

Slender beams in active bending allow the realization of intriguing structures, with innovative uses in the construction of frames and facades, furniture and product design, and even machine engineering, as seen in Fig. 4.2. Despite a firm grounding in history, such as its centuries-old role in the construction of Turkmen tents, the aesthetic of curves that emerge from pure bending still fascinates designers, architects, and mathematicians today [LAGK13].

The primary allure of bending as part of the formation process is its economic advantage: Curved members can be cut from flat, inexpensive sheets of elastic material such as plywood, which allows for easy transportation, and assembly on-site. This has the promise of reducing the need for an individual mold per unique structural member. Active bending also brings with it a singular design and form-finding challenge: The design space of shapes that can be physically realized is dictated by the underlying mechanics of slender beams. Insights about the geometric structure of this design space are therefore of practical value, because they can guide the intuition of the designer towards feasible designs. As we will show, they also open

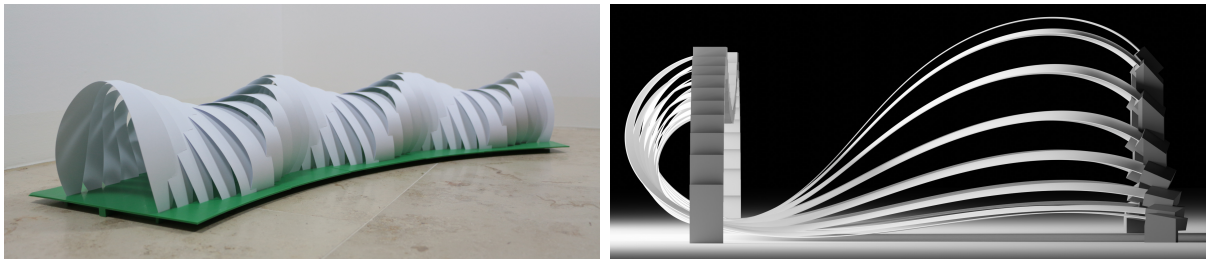


Figure 4.1: **Pavilion and Carpet**. *Left*: Photograph of a physical pavilion model, constructed by bending inexpensive strips of plane cardboard. The flat outline of all strips is determined computationally within 0.3 seconds, in order to exactly match the design under their own weight. *Right*: Rendering of active-bending design for a free-form structure. The beam stiffness profiles adapt to design edits in real time, and a form-finding algorithm improves manufacturability.

Figure 4.2: **Examples of Active Bending**. *Left to right, top to bottom:* ICD/ITKE Research Pavilion 2010 [FM11]. TR11 lamp by Tom Rossau. PS1 Loop, Boston, by Höweler + Yoon Architecture. Bentwood type "yourte" (CC BY 2.0 Jean-Pierre Dalbéra). Morphing rotor blade controlled by a servomotor (CC BY 3.0 DLR). Arc lamp [CMM18].

the door to extremely efficient optimization algorithms that further support the form-finding process.

The design space of slender beams in plane bending is related to the *Elastica problem*, a classical question in mechanics: Given a straight, slender structural element with constant stiffness, enumerate all its static equilibria in the plane. Originally posed by Bernoulli, a full enumeration as a one-parameter family was given by Saalschütz in terms of elliptic functions in 1880. This result has historically influenced the use of mechanical splines in design, but the assumption of constant stiffness limits the space of equilibrium curves attained within the Elastica framework.

Digital fabrication methods simplify the manufacture of structural elements with spatially varying cross-sectional profiles, and thus variable stiffness. Yet, we are unaware of any work that enumerates or classifies all shapes that can be attained as equilibrium states of variable-stiffness elastic curves. This suggests two questions: Is it possible to characterize all plane curves that can be attained as static equilibria of slender beams? And how can we construct the geometry of a beam that realizes a given equilibrium curve?

In this work, we give positive answers to both questions. This results in a geometric characterization of all variable-stiffness elastic plane curves that is both intuitive and mathematically sharp. We also show a method to compute the stiffness profile required to realize a given elastic curve, which is globally optimal with respect to a convenient fabricability objective. These computations only take a fraction of a second, which makes them ideal for interactive design, where they provide immediate feedback about the practicality of a structure. At the same time, they can be integrated in iterative form-finding algorithms that suggest design trade-offs to the user at interactive speeds. We explore both use cases in this work.

For many real-world applications, the dead load and stability of a member play an important role in the design process. This is because a beam may deform considerably under its own weight, and an unstable element will even change its shape completely by snapping into a different equilibrium. These effects are not part of the classical Elastica model, but we show that the dead load can be incorporated into our construction method with virtually no penalty to computation time. The stability problem is more difficult and, in our experience, beyond the grasp of human intuition alone: It is often impossible to tell by visual inspection whether

a given curve is stable, or even whether there could be a similar curve that is stable. We show that our geometric characterization of elastic curves is the key ingredient in designing an algorithm that can modify an unstable curve to yield a similar—and sometimes visually indistinguishable—stable curve.

Our contributions impact design with active bending in three ways. First, the geometric characterization of elastic curves informs a designer about the kinds of shapes that are physically possible within active-bending structures. This prevents infeasible designs even in the first phase of conception. Second, our gravity-aware construction method for stiffness profiles gives instant feedback about the practicality of a design. Namely, it tells us whether it can be realized with a given material and fabrication method. Third, we enable optimization routines that automatically improve the fabricability of a design with small changes to form—a process that takes under one second per beam in all of our examples.

4.2 Overview

We begin our considerations with the variable-stiffness version of the classical Elastica problem: Which plane curves can be obtained as equilibria of slender beams? In Section 4.3, we show that the main geometric condition is the existence of a line that intersects a curve in its inflection points, and nowhere else. Building on this, we describe a construction process for a stiffness profile that realizes a given equilibrium curve, and which is optimal—in a well-defined sense—for fabrication. Section 4.4 proposes a modification that lets us incorporate the dead load of a member into this construction process. In Section 4.5, we turn to the stability of elastic curves and devise an algorithm that turns an unstable curve into a stable one, sometimes by imperceptibly small modifications. We will see that our solution to the variable-stiffness Elastica problem provides the key insight to enable this application. Section 4.6 gives details about the discretization and implementation of the concepts discussed thus far. In Section 4.7, we describe and validate the fabrication process behind the physical active-bending models that we present in Section 4.8, along with rendered examples and quantitative data.

In particular, we consider these our main technical contributions:

- A geometric characterization of all plane elastic curves (Sections 4.3.2, 4.3.3, Appendix A.1);
- A fast and fabrication-friendly construction algorithm for stiffness profiles of elastic curves (Section 4.3.4);
- An extension of this algorithm that accounts for the dead load of the structural member (Sections 4.4.1–4.4.3);
- A derivation of the adjoint equations for the Jacobi stability criterion, and their use in stability optimization of elastic curves (Sections 4.5.2–4.5.4, Appendix A.3);
- Fabrication methods for realizing stiffness profiles as elastic strips (Section 4.7.1).

4.3 Equilibrium Curves

This section treats the most straight-forward adaptation of the Elastica problem to our setting: Characterize the set of plane curves that occur as static equilibria of straight elastic rods. The

crucial difference to the original Elastica problem is that we explicitly allow the stiffness of a rod to vary across its length.

We show that this set of equilibrium curves is characterized by an intuitive geometric property having to do with collinearity of inflection points. In addition, the characterization is computationally convenient and gives rise to an algorithm that finds the "best"—in a well-defined sense—geometry of a rod that matches the desired equilibrium curve.

We first introduce the mathematical model for kinematic elastic rods and arrive at an equilibrium equation. Then, we characterize the set of all curves satisfying this equation with a suitable constitutive law. Finally, we present a linear program that finds the optimal stiffness profile for a desired equilibrium shape, with respect to a manufacturability objective.

4.3.1 Mathematical Model

Preliminaries The deformed state of an inextensible elastic rod in the plane is modeled as a curve $\gamma : [0, l] \to \mathbb{R}^2$, passing through the centerline of the rod. For notational convenience, we assume an arc-length parametrization, i.e., $\|\gamma'\| \equiv 1$. But for a computer implementation of algorithms presented in this paper, any regular parametrization will suffice. We only assume $\gamma \in C^2$, so the signed curvature $\kappa = \det(\gamma', \gamma'')$ is continuous. In addition, denote by $\alpha \in C^1$ the turning angle of γ, so $\gamma' = (\cos\alpha, \sin\alpha)^t$ and $\alpha' = \kappa$. The notation used throughout this section is summarized in Fig. 4.3.

The resistance of a rod to bending is given by a stiffness function $K : [0, l] \to \mathbb{R}_{>0}$, which determines the ratio between curvature and moment of force at any point. E.g., if a linearly elastic rod has a rectangular cross section with width w and thickness h at $s \in [0, l]$, its stiffness will be $K(s) = \frac{1}{12} E w h^3$, where E is the Young's modulus of the material. Most examples presented in this paper will have constant thickness h, so the width w scales linearly with K. Real materials cannot exhibit stiffnesses that are arbitrarily low or high, which motivates:

Definition 1. Let $K : [0, l] \to \mathbb{R}_{>0}$ such that there exist $c, C \in \mathbb{R}$ with $0 < c \leq K(s) \leq C$ for all $s \in [0, l]$. Then K is called *admissible*.

We will study the design space offered by admissible stiffness functions, because they correspond to elastic rods that we can manufacture in the real world. Note that the existence of a positive lower bound c is a strictly stronger requirement than $K > 0$ and ensures that the ratio between $\sup K$ and $\inf K$ is finite.

Equilibrium Equation Equilibrium configurations of an elastic rod are characterized by extremals of the bending energy $\int \frac{1}{2} K \kappa^2$, subject to boundary conditions and constraints that reflect how the ends of the rod are fixed. We will assume *kinematic* rods, i.e., rods in which $\alpha(0), \gamma(0)$ and $\alpha(l), \gamma(l)$ are all constrained. This leads to a variational problem with Dirichlet boundary conditions and two integral constraints, and we are looking for extremals of

$$W[\alpha] = \int_0^l \frac{1}{2} K (\alpha')^2 \quad \text{s.t.} \quad \begin{matrix} \alpha(0) = \alpha_0, \\ \alpha(l) = \alpha_l, \end{matrix} \quad \text{and} \quad \int_0^l \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} = \gamma_l. \tag{4.1}$$

The endpoint $\gamma(0)$ is assumed to coincide with the origin, and the endpoint $\gamma(l) = \int_0^l \gamma' = \int_0^l \left(\begin{smallmatrix} \cos\alpha \\ \sin\alpha \end{smallmatrix}\right)$ is constrained to lie at γ_l.

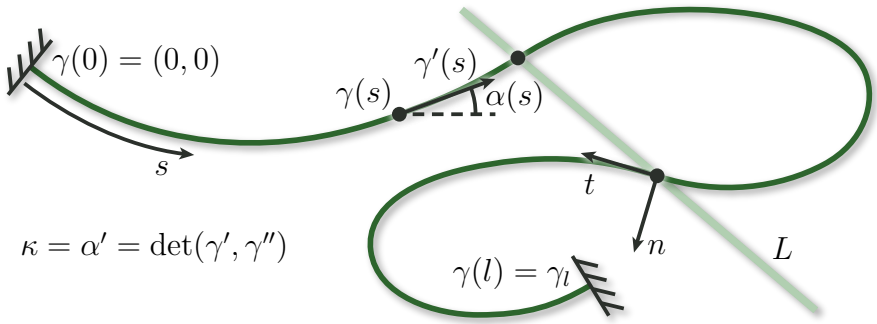


Figure 4.3: **Curve Description.** Plane curve γ with arc-length parameter s, turning angle α, signed curvature κ, and length l. The endpoints are the origin and γ_l. The pair (t, n) forms a right-handed coordinate system adapted to γ at an inflection point, and L is a line incident to all inflection points.

The constrained Euler–Lagrange equation of this problem is

$$-(K\kappa)' + \langle \lambda, R\gamma' \rangle = 0, \tag{4.2}$$

where $R = \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$, and the Lagrange multiplier $\lambda \in \mathbb{R}^2$ needs to be chosen to satisfy the integral constraints. Next, we rename $\lambda = Rb$, which simplifies $\langle \lambda, R\gamma' \rangle = \langle b, \gamma' \rangle$, and integrate the equation through. This adds an integration constant $a \in \mathbb{R}$ and yields the moment equilibrium equation

$$K\kappa = a + \langle b, \gamma \rangle. \tag{4.3}$$

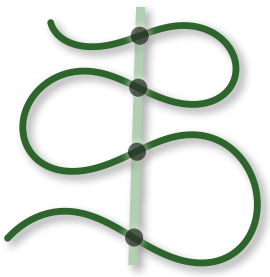
We characterize curves that satisfy this equation by:

Definition 2. Let $\gamma \in C^2([0, l]; \mathbb{R}^2)$ be an arc-length parametrized plane curve with signed curvature κ. If there exist $a \in \mathbb{R}$, $b \in \mathbb{R}^2$, and an admissible stiffness function K such that $K\kappa = a + \langle b, \gamma \rangle$, then γ is called an *equilibrium curve*.

This definition is chosen so a curve γ is an equilibrium curve if and only if it is possible to manufacture a straight elastic rod that has γ as an equilibrium shape when kinematic boundary conditions are applied. In the next section, we show how the set of all equilibrium curves can be characterized geometrically. This results in a description of the design space of plane kinematic rods that is both intuitive and mathematically sharp.

4.3.2 Characterization of Equilibrium Curves

If γ is a line segment, the rod is undeformed, and Eq. 4.3 is trivially solved by $a = 0$, $b = 0$, and K arbitrary. The theorem below treats the more interesting case, in which the rod undergoes bending.
It shows that the main geometric condition for γ to be an equilibrium curve is the existence of a line that intersects γ exactly in its inflection points (see inset). The only technical assumption we make, apart from $\gamma \in C^2$, is that the number of inflection points be finite, which is a natural expectation in the context of design.



Before we state the theorem, we introduce one more definition:

Definition 3. A function $f : \mathbb{R} \to \mathbb{R}$ with $f(x_0) = 0$ is called secant-bounded at x_0 if there exist $\varepsilon > 0$ and $m, M \in \mathbb{R}$ with either $m, M > 0$ or $m, M < 0$ such that, for all $h \in (-\varepsilon, \varepsilon)$,

$$\min\{mh, Mh\} \leq f(x_0 + h) \leq \max\{mh, Mh\}.$$

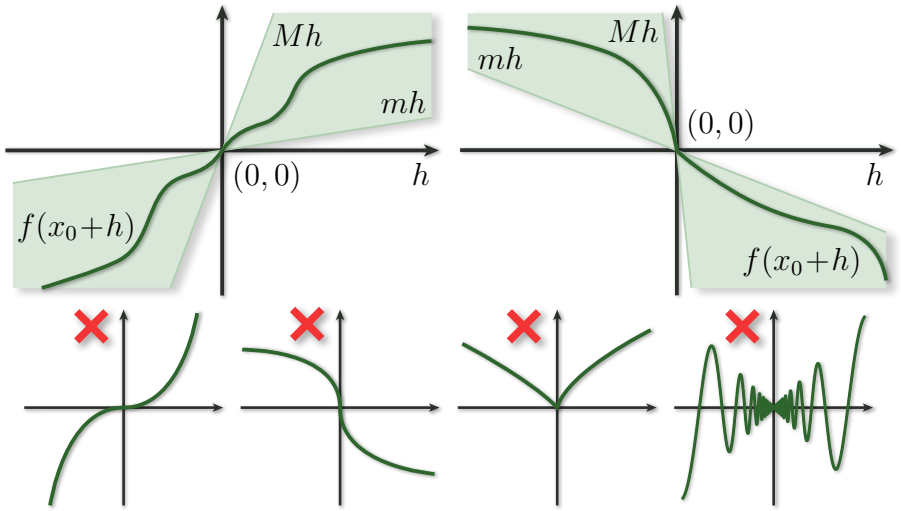


Figure 4.4: **Secant-boundedness.** *Top:* A function $f : \mathbb{R} \to \mathbb{R}$ with $f(x_0) = 0$ is secant-bounded at x_0 if its graph, restricted to some interval $(x_0 - \varepsilon, x_0 + \varepsilon)$, is contained in a double cone that is bounded by two linear functions mh and Mh, where either $m, M > 0$ *(left)* or $m, M < 0$ *(right)*. *Bottom:* Examples of functions that fail to be secant-bounded at the origin.

Fig. 4.4 illustrates secant-boundedness and gives examples of functions that do not have this property.[1] Now we are ready to state our main theoretical result, which is proved in Appendix A.1:

Theorem 1. *Let $\gamma \in C^2([0, l]; \mathbb{R}^2)$ be an arc-length parametrized plane curve with signed curvature $\kappa = \det(\gamma', \gamma'')$, and the set of zero-curvature parameters $S_0 := \{s \in [0, l] : \kappa(s) = 0\}$ finite. Then, γ is an equilibrium curve if and only if*

1. *there exists a line L that intersects γ exactly in its zero-curvature points, and that is not tangent to γ in any of these intersections;*
2. *κ is secant-bounded at every $s_0 \in S_0$.*

Remark 1. The secant-boundedness of κ on S_0 implies that κ actually changes sign at every root; but it is even stronger, as seen in Fig. 4.4 (bottom). In case $\kappa \in C^1$, the condition reduces to $\kappa'(s_0) \neq 0$.

Remark 2. In the context of design, condition (2) may seem overly technical to be of practical relevance. However, in Section 4.5, we will see that this condition is computationally significant because it causes an algorithm to fail unless explicitly enforced.

4.3.3 Properties of Equilibrium Curves

The geometric characterization given above is intuitive enough that we can usually tell by inspection whether a curve is an equilibrium curve. The main condition is that all inflection points lie on a line which does not cross the curve in non-inflectional points. In particular, this means that every curve without inflections is an equilibrium curve. For curves with at least one

[1] Note that secant-boundedness is weaker than being locally bi-Lipschitz at x_0, which is defined as follows: there exist $\varepsilon, L > 0$ such that for all $x, y \in (x_0 - \varepsilon, x_0 + \varepsilon)$, it holds that $|x - y|/L \leq |f(x) - f(y)| \leq L|x - y|$. In particular, locally bi-Lipschitz functions are locally injective—secant-bounded functions need not be.

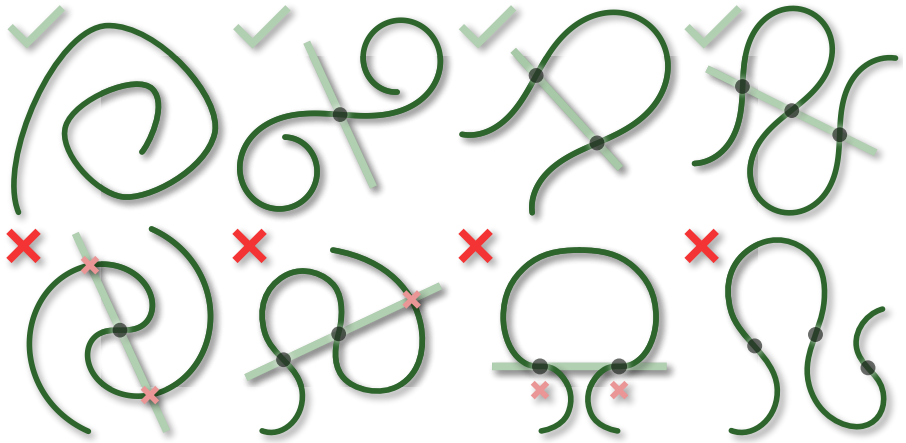


Figure 4.5: **(Non-)Equilibrium Curves.** *Top:* Four equilibrium curves, from a non-inflectional curve *(left)* to a curve with three inflections *(right)*. *Bottom:* Four curves that are not equilibrium curves because *(from left to right)* every line through the inflection intersects the curve in further points; the unique inflection line intersects the curve in a non-inflectional point; the inflection line is tangent to the curve; the inflection points are not collinear.

inflection, it depends on the global shape whether it has the equilibrium property, as shown by example in Fig. 4.5. We can infer some more properties that are useful in a design context:

Locality. Every curve that satisfies the technical requirements of Theorem 1 is locally an equilibrium curve, i.e., it can be split into a finite number of equilibrium curves. Thus, every curve can be realized with an elastic rod if we introduce additional fixed points.

Projective invariance. The class of equilibrium curves is closed under projective transformations. This holds because collinearity of inflection points is a projective invariant. It is thus possible to create new equilibrium curves by editing an existing curve with projective transformation tools.

Smoothness of K. The smoothness of the stiffness function will affect the visual appearance of a manufactured rod, and may thus be of interest. If $\gamma \in C^r$, we infer from $K = \frac{a+\langle b,\gamma\rangle}{\kappa}$ that K is at least C^{r-2} away from inflection points, and at least C^{r-3} at inflection points. E.g., to ensure that K is continuous in a design with inflections, one may use quartic splines, so $\gamma \in C^3$. Without inflections, cubic splines suffice.

The applicability of Theorem 1 can be expanded by considering the following variants, which can be proved in the same way as the original statement:

Boundary conditions. Theorem 1 assumes kinematic boundary conditions to maximize the design space, but we can account for the effect of removing positional or angle constraints by imposing additional requirements on γ and L. If the endpoint constraint on x (y) is removed, this constrains L to be vertical (horizontal). If both are removed, γ must not have inflections. If the angle constraint at either endpoint is removed, the curvature of γ needs to vanish at this endpoint, which in turn requires L to intersect it.

Fixed natural curvature. By substituting every occurrence of "κ" in Theorem 1 with "$\kappa - \kappa_0$", we can also characterize the equilibrium curves of plane rods with fixed natural curvature κ_0.

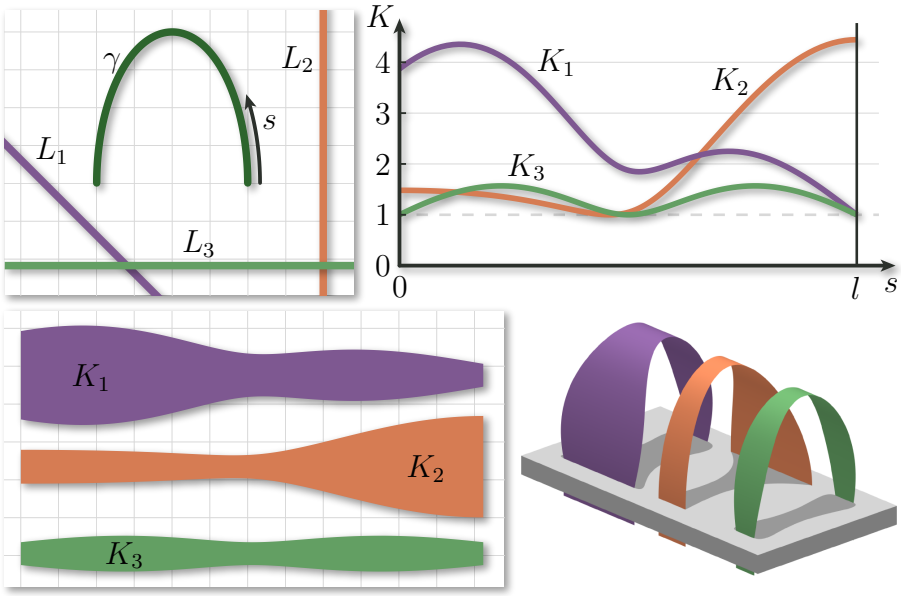


Figure 4.6: **Stiffness Profiles.** *Top left:* An elliptical arc γ and three lines representing viable choices of a, b. *Top right:* Stiffness profiles induced by the lines; K_3 was computed using Eq. 4.4 and minimizes R globally. *Bottom left:* Outlines of elastic strips realizing the stiffness profiles. *Bottom right:* Deformed rods with boundary conditions applied, yielding identical shapes.

4.3.4 Computation of Stiffness Profiles

To synthesize the geometry of a rod, it suffices to prescribe its stiffness at every $s \in [0, l]$. The stiffness can then be transformed into a cross section, e.g., a rectangular cross section with width w and thickness h, such that K is proportional to wh^3 everywhere.

The stiffness function of an equilibrium curve is not generally unique, because it depends on the choice of $a \in \mathbb{R}$ and $b \in \mathbb{R}^2$ via $K = \frac{a+\langle b,\gamma\rangle}{\kappa}$. For curves with two or more inflections, L is unique and thus defines a and b up to scalar multiples. But for curves with zero or one inflection, additional degrees of freedom remain. This raises the question of what the "best" stiffness function is, and how it can be computed. In a fabrication setting, one is typically limited by the range of stiffnesses that can be reliably achieved within a single structural element. Thus, in our view, the best stiffness function is the one that minimizes the max-to-min stiffness ratio $R[K] := \frac{\sup K}{\inf K}$.

To find the global minimizer of R, note that R is naturally scale-invariant: If K is an admissible stiffness for γ, so is θK for any $\theta > 0$. Likewise, if K is a minimizer of R, so is θK. This means that we can eliminate scalar multiples without losing any solutions, for example by imposing the affine constraint $\inf K = 1$. In the affine subspace defined by this constraint, the objective simplifies to $R[K] = \sup K$, which shows that the problem of minimizing R is in fact linear, and can be discretized using a linear program.

To do this, let $S_0 = \left\{s_1^{\text{infl}}, \ldots, s_m^{\text{infl}}\right\}$, and $0 = s_1, \ldots, s_n = l$ a sampling of $[0, l]$ that does not include any of the inflection points in S_0. The sampling should be dense enough so $\max\{K(s_1), \ldots, K(s_n)\}$ approximates $\sup K$ well. Then, solve the following linear program

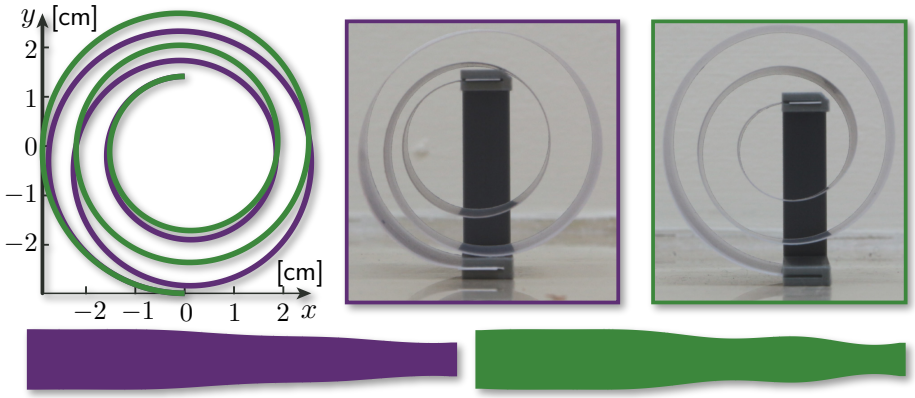


Figure 4.7: **Spiral under Gravity.** *Top:* Target curve (green) and predicted shape if gravity is neglected in the linear program (purple); physical models, computed with Eq. 4.4 *(center)*, and with Eq. 4.6 *(right)*. *Bottom:* Shapes of the elastic strips cut from cardboard.

in the variables $a, R \in \mathbb{R}$ and $b \in \mathbb{R}^2$:

$$\begin{aligned} &\text{minimize } R, \\ &\text{subject to } 1 \leq \frac{a + \langle b, \gamma(s_i)\rangle}{\kappa(s_i)} \leq R, && i = 1, \ldots, n, \\ &\qquad 0 = a + \langle b, \gamma(s_i^{\text{infl}})\rangle, && i = 1, \ldots, m. \end{aligned} \tag{4.4}$$

The fraction appearing in the linear program equals $K(s_i)$, and $\min_i K(s_i) = 1$ is implicitly enforced by the inequality constraints $1 \leq K(s_i)$. The equality constraints ensure that L intersects all inflection points of the curve. Furthermore, it is guaranteed that L does not intersect the curve in non-inflectional points, because this would result in a negative stiffness value at a sample adjacent to the intersection. With only four variables and $2n+m$ constraints, this linear program is very small and can be solved almost instantaneously for arbitrary curves.

Fig. 4.6 shows how the stiffness profile for a curve varies with the choice of a and b. An arbitrary choice may result in unwieldy rod geometries such as K_1 and K_2. The profile K_3, which is the global minimizer of $R[K]$, was found by solving the linear program.

4.4 Equilibrium Curves Under Gravity

The theory presented in the previous section suffices to create designs on a scale where the effect of gravity is negligible. For applications in which the dead load of a beam significantly affects the equilibrium shape, like the one shown in Fig. 4.7, we need to model gravity explicitly. Although we cannot use Theorem 1 to determine feasibility in this case, we show that it is possible to adapt the computation of stiffness profiles to account for gravity exactly.

The main finding is that the problem of determining feasibility and finding the "best" stiffness profile under gravity remains linear if the thickness of the material is known a priori. Furthermore, the complexity of the linear program used to solve it only increases marginally, and solutions can still be found near instantaneously.

4.4.1 Mathematical Model

The gravity potential of a body $V \subset \mathbb{R}^3$ with density ρ in the earth's gravitational field is given by $U = \int_V \rho(x)\langle x, g\rangle \mathrm{d}x$, with g the gravitational acceleration vector. For an

elastic strip with constant thickness h and variable width w, we can write this integral as $U = \int_0^l \rho h w(s) \langle \gamma(s), g \rangle \mathrm{d}s$. We can also express the stiffness in terms of these quantities, which yields $K = \frac{1}{12} E w h^3$, with E the Young's modulus of the material. Substituting w for K and summing gravity potential and bending energy gives the functional

$$W[\alpha] = \int_0^l K \left(\frac{1}{2}\kappa^2 + \langle \gamma, e \rangle \right) \quad \text{with } e = \frac{1}{\beta l^3} \cdot \frac{g}{\|g\|}, \quad \beta = \frac{E h^2}{12 \rho \|g\| l^3},$$

where β is the standard gravito-bending parameter [AP10, 4.1.2]. We estimate this parameter by cutting a rectangular strip of a material, and measuring the displacement in a cantilever experiment. Then, we fit the parameter so the minimizer of W matches the observed displacement. For 200 gsm cardboard, we found $\beta l^3 = 8.86 \cdot 10^{-4} \mathrm{m}^3$, which we use for all examples.

The appearance of $\gamma(s) = \int_0^s \left(\begin{smallmatrix} \cos\alpha \\ \sin\alpha \end{smallmatrix}\right)$ in the energy means that W depends on α via a double integral. To find the Euler–Lagrange equation of a functional like this, we compute the variational derivative with respect to α, and then test against a delta-distribution centered at s, as detailed in Appendix A.2. This gives

$$-(K\kappa)'(s) + \langle b, \gamma'(s) \rangle + \langle R^t e, \gamma'(s) \rangle \int_s^l K(t)\,\mathrm{d}t = 0,$$

where $R = \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$. This equation can be integrated to yield the moment equilibrium equation

$$\begin{aligned} -K(s)\kappa(s) + \langle b, \gamma(s) \rangle + \langle R^t e, \gamma(s) \rangle \int_s^l K(t)\,\mathrm{d}t \\ - \int_s^l \langle R^t e, \gamma(t) \rangle K(t)\,\mathrm{d}t + a = 0. \end{aligned} \tag{4.5}$$

A comparison with Eq. 4.3 shows that two new summands involving integrals have appeared due to the effect of gravity. Below, we show how to account for them in the computation of stiffness profiles.

4.4.2 Computation of Stiffness Profiles

In Eq. 4.5, the stiffness K appears in three different forms: evaluated at s, integrated from s to l, and once more integrated against $\langle R^t e, \gamma \rangle$. All three appearances are linear in K, so we can still cast this as a linear program.

To do this, let $0 = s_1, \ldots, s_n = l$ be a dense sampling of $[0, l]$. Associate with the samples a set of weights $w_1, \ldots, w_n$ based on the Voronoi lengths of the samples along the curve, so $\int_{s_k}^{s_n} f \approx \sum_{i=k}^n w_i f(s_i)$. In addition to $a, R \in \mathbb{R}$ and $b \in \mathbb{R}^2$, the linear program has auxiliary variables for K, and for the two types of integrals that appear in Eq. 4.5, call them F and G. Then, the program is given by

$$\begin{aligned} &\text{minimize } R, \\ &\text{subject to } 1 \le K_i \le R, && i = 1, \ldots, n, \\ &\qquad 0 = -K_i \kappa(s_i) + \langle b, \gamma(s_i) \rangle \\ &\qquad\quad + \langle R^t e, \gamma(s_i) \rangle F_i - G_i + a, && i = 1, \ldots, n, \\ &\qquad F_i = F_{i+1} + w_i K_i, && i = 1, \ldots, n, \\ &\qquad G_i = G_{i+1} + w_i \langle R^t e, \gamma(s_i) \rangle K_i, && i = 1, \ldots, n, \end{aligned} \tag{4.6}$$

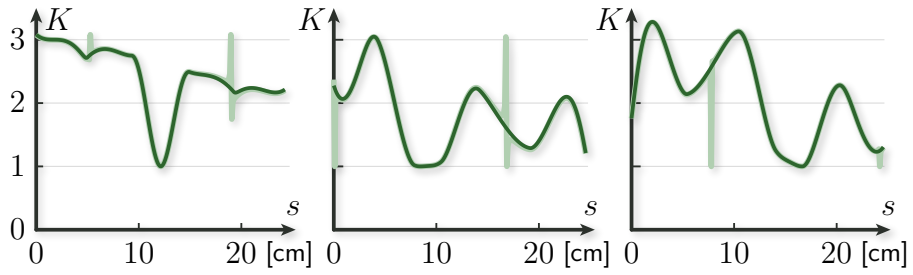


Figure 4.8: **Inflectional Curves under Gravity.** Stiffness functions after solving Eq. 4.6 (light green) and after solving the modified linear program (dark green) for curves taken from the flower pot example, see Fig. 4.19.

where we set $F_{n+1} = 0 = G_{n+1}$. The last two lines constrain $F_i \approx \int_{s_i}^{l} K$ and $G_i \approx \int_{s_i}^{l} \langle R^t e, \gamma \rangle K$. In total, the program has $3n+4$ variables, $2n$ inequality constraints, and $3n$ equality constraints. It is possible to eliminate the variables F_i, G_i and their defining equality constraints by substitution. But even in its stated form, the program can be solved to optimality within a few milliseconds for any curve.

Fig. 4.7 shows a challenging example, in which precise estimation of the gravity parameter e is paramount. The influence on the stiffness profile is subtle, but shape reproduction is significantly improved: Neglecting gravity yields a model that sags under gravity and collides with the support structure; taking gravity into account explicitly results in a uniform spiral that is only supported at the endpoints.

4.4.3 Inflectional Curves

If admissible stiffness profiles exist for a given curve, Eq. 4.6 always finds the one that minimizes R. However, this minimizer is sometimes surprisingly noisy near inflection points, which is a byproduct of curve discretization. In particular, K tends to have spikes like the ones shown in Fig. 4.8 (light green), which allow for a small decrease in R compared to a more preferable, smooth solution.[2] To remove these spikes, and still retain a solution to Eq. 4.5, we solve a second linear program that is obtained from Eq. 4.6 by two modifications:

- Replace the objective function by a discretization of $\mathrm{V}(K') = \int_0^l |K''|$, the total variation of K'.
- Add the constraint $R \leq R_1 \cdot (1+\varepsilon)$, where R_1 is the optimal value obtained from solving Eq. 4.6. We always set $\varepsilon = 0.01$.

The new objective function favors solutions without noise, and the original optimum is retained up to a margin of ε. This new linear program is feasible whenever the original problem is feasible, because the optimal point of Eq. 4.6 is feasible in the new program. We have used this two-phase optimization in every example that treats inflectional curves under gravity, and it has yielded completely noise-free results in every case, like the examples in Fig. 4.8.

[2] Note that this only happens when gravity is taken into account. In contrast, solutions to Eq. 4.4 are always noise-free, regardless of inflections, because the inflection line is sufficiently constrained by the equation $0 = a + \langle b, \gamma(s_i^{\text{infl}}) \rangle$.

4.5 Stability of Equilibrium Curves

A solution to a variational problem is said to be stable if it is a local minimum of the energy functional. Unstable solutions are saddle points, which means that there is a perturbation of the solution, compatible with boundary conditions and constraints, that attains lower energy. For a physical system, this means that force equilibrium holds at a saddle point, but any disturbance will cause it to become dynamic, rendering it useless for most applications.

Overview From visual inspection alone, it is far from obvious whether a specific equilibrium curve will be stable, or even whether there could be a *similar* equilibrium curve that is stable. The remainder of this section describes an optimization algorithm that, given an unstable equilibrium curve, finds a similar equilibrium curve which is stable. We believe this is the first algorithm of this type.

Our method is based on the theory of isoperimetric conjugate points, which we review below. Then, we present our idea for stability recovery, based on applying the adjoint method to the isoperimetric Jacobi equations. Before discussing our final algorithm, we outline why a simpler and more obvious version of the algorithm fails. Theorem 1 plays a crucial role in formulating the method, so it is currently limited to applications in which gravity is negligible.

4.5.1 Stability Criteria

There are two essential procedures for analyzing the stability of a solution to an optimization problem: the eigenvalue test and the Jacobi test. The eigenvalue test is used more frequently in practice, but we will argue that the Jacobi test is better suited to the problem at hand and leads to an elegant optimization algorithm at a low computational cost.

The Eigenvalue Condition A sufficient criterion for stability is the positivity of all eigenvalues of the energy Hessian. For constrained problems, there exists a similar criterion involving eigenvalues of directions constrained to the orthogonal subspace of constraint tangents. This method is very general, but it does not take advantage of the structure inherited from continuous variational problems.

There are two obstacles to using the eigenvalue method for optimizing stability of kinematic elastic curves. First, such optimizations are iterative, and require solving a constrained eigenvalue problem at every step. These computations are very costly because standard numerical packages such as LAPACK do not support solution procedures that preserve sparsity for this case. Second, unlike the unconstrained case [VdAMM07], we have found no reference on computing derivatives for constrained eigenvalue problems.

The Jacobi Condition A different stability criterion can be derived directly from the continuous variational problem, and then discretized. This condition, named after Jacobi, can be checked by integrating simple initial-value problems on $[0, l]$. The result regarding stability is equivalent to that of the eigenvalue test, despite a significantly lower computational cost, and a complexity that is only linear in the number of samples. In addition, the criterion we use for establishing stability is differentiable via the adjoint method, and not subject to singularities that might be present in the eigenvalue structure. Below, we review a version of the Jacobi condition that allows for multiple equality constraints [MRM98].

4.5.2 Isoperimetric Conjugate Points

General Theory In this section, we summarize the sufficient stability criterion for constrained variational problems based on conjugate points, omitting derivations. We start with a sufficiently regular functional $J[u] = \int_0^l f(x, u(x), u'(x))\,\mathrm{d}x$ subject to Dirichlet b.c., and integral constraints $\int_0^l g_i(u(x))\,\mathrm{d}x = c_i$, for $i = 1, \dots, p$. Then, the Lagrangian of the problem is given by $\mathcal{L}[u] = \int_0^l f^*(x, u, u')\,\mathrm{d}x$, where $f^* = f + \sum_i \lambda_i g_i$. The Lagrangian can be developed into a second-order Taylor series as

$$\mathcal{L}[u+h] = \mathcal{L}[u] + \int_0^l \left(f^*_u - \frac{\mathrm{d}}{\mathrm{d}x} f^*_{u'} \right) h + \frac{1}{2} \int_0^l \left(P h'^2 + Q h^2 \right) + o(\|h\|_w^2),$$

$$\text{with} \quad P = f^*_{u'u'}, \quad Q = f^*_{uu} - \frac{\mathrm{d}}{\mathrm{d}x} f^*_{uu'}, \quad \|h\|_w^2 = \|h\|_{L^2}^2 + \|h'\|_{L^2}^2.$$

In analogy to minimization problems with finite dimension, J has a constrained minimum at u if the first-order term in the expansion of $\mathcal{L}$ vanishes, and the second-order term is positive-definite within the orthogonal subspace of constraint tangents. The first-order condition is exactly the constrained Euler–Lagrange equation. Positive-definiteness of the second-order term is guaranteed if the so-called Legendre and Jacobi conditions are satisfied. The Legendre condition is met if $P > 0$ everywhere.

For the Jacobi condition, we need the concept of isoperimetric conjugate points. First, let ζ be the solution to the *Jacobi equation*,

$$-(P\zeta')' + Q\zeta = 0 \quad \text{s.t.} \quad \zeta(0) = 0,\ \zeta'(0) = 1,$$

and let η_i, for $i = 1, \dots, p$, be the solutions to

$$-(P\eta_i')' + Q\eta_i = T_i \quad \text{s.t.} \quad \eta_i(0) = 0,\ \eta_i'(0) = 1,$$

where $T_i := \mathrm{d}g_i/\mathrm{d}u$ are the constraint tangents. Next, we define the running integrals

$$M_i(x) = \int_0^x T_i \zeta \quad \text{and} \quad N_{ij}(x) = \int_0^x T_i \eta_j \quad \text{for} \quad i, j = 1, \dots, p.$$

This gives the entries of the *constrained stability matrix*,

$$Z(x) = \begin{pmatrix} \zeta(x) & \eta_1(x) & \cdots & \eta_p(x) \\ M_1(x) & N_{11}(x) & \cdots & N_{1p}(x) \\ \vdots & \vdots & \ddots & \vdots \\ M_p(x) & N_{p1}(x) & \cdots & N_{pp}(x) \end{pmatrix}.$$

Finally, let $\mathcal{Z}(x) := \det Z(x)$. A point $\sigma \in \mathbb{R}$ with $\mathcal{Z}(\sigma) = 0$ is called an *(isoperimetric) conjugate point*. The Jacobi condition is satisfied if there is no conjugate point in $(0, l]$.

In summary: If, at u, the Euler–Lagrange equation is satisfied, and $P(x) > 0$ for all $x \in [0, l]$, and there is no conjugate point in $(0, l]$, then u is a minimizer.

Elastic Rods Now, we apply this theory to kinematic elastic rods, to determine whether an equilibrium curve γ is stable. Stability is mostly an issue for curves with at least two inflection points, so K and λ (see Eq. 4.2) are uniquely determined by γ up to positive multiples. Using the variable names from above, we have

$$g_1(\alpha) = \cos\alpha,\ g_2(\alpha) = \sin\alpha, \quad T_1(\alpha) = -\sin\alpha,\ T_2(\alpha) = \cos\alpha,$$
$$P = K, \quad Q = -\lambda_1 \cos\alpha - \lambda_2 \sin\alpha = -\langle \lambda, \gamma' \rangle,$$

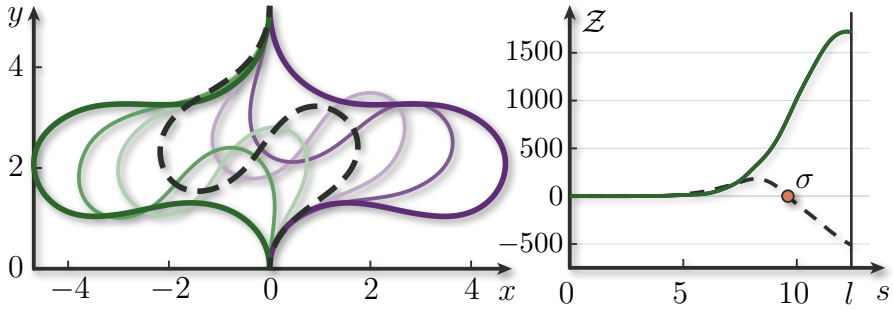


Figure 4.9: **Stability and Conjugate Points.** *Left:* Unstable equilibrium (dashed), two stable equilibria (green, purple), and in-between non-equilibrium states of the same rod. *Right:* Stability determinant $\mathcal{Z}(s)$ for unstable and stable equilibria, with σ marked (orange).

so the Legendre condition $P = K > 0$ is always satisfied. The Jacobi equation reads

$$-(K\zeta')' - \langle \lambda, \gamma' \rangle \zeta = 0 \quad \text{s.t.} \quad \zeta(0) = 0,\ \zeta'(0) = 1, \tag{4.7}$$

and is readily solved by integrating it from $s = 0$ to l. The functions η_1 and η_2 are determined by solving with the right-hand sides T_1 and T_2, respectively. Finally, M_1, M_2, N_{11}, N_{12}, N_{21}, and N_{22} are obtained by integrating the solutions against the constraint tangents. This gives all entries in the 3-by-3 stability matrix Z, whose determinant $\mathcal{Z}$ can be checked for zero-crossings on $(0, l]$. If the sign of $\mathcal{Z}$ remains constant on this interval, then γ is a stable equilibrium curve, and otherwise it is unstable. Fig. 4.9 shows how $\mathcal{Z}$ differs between an unstable and a stable equilibrium of the same elastic rod.

If an equilibrium curve is unstable, the location of the first isoperimetric conjugate point $\sigma \in (0, l]$ indicates how close the curve is to being stable—if σ is close to l, then the curve is "almost" stable. (This is analogous to the magnitude of the lowest negative eigenvalue.) Our goal is to make a curve stable by pushing σ towards l, and finally out of the interval $(0, l]$, while modifying the curve as little as possible.

Adjoint Method The function $\mathcal{Z}(s)$ depends on K via the Euler–Lagrange and Jacobi equations, so we write it as $\mathcal{Z}(s, K)$. A conjugate point is implicitly defined via $\mathcal{Z}(\sigma, K) = 0$. We can apply the implicit function theorem to find

$$-\partial\mathcal{Z}/\partial s|_{(\sigma,K)} \cdot \mathrm{d}\sigma/\mathrm{d}K = \partial\mathcal{Z}/\partial K|_{(\sigma,K)}.$$

If we can numerically compute $\mathrm{d}\sigma/\mathrm{d}K$, this gives us a way to apply gradient-based optimization on K in order to push σ towards l. But the equation shows that $\mathrm{d}\sigma/\mathrm{d}K$ coincides with $\partial\mathcal{Z}/\partial K|_{(\sigma,K)}$ up to a scaling factor $-\partial\mathcal{Z}/\partial s|_{(\sigma,K)}$, which can be shown to always be positive. Because $\mathcal{Z}$ depends on K via a series of differential equations and integrals, we need to derive its adjoint equations in order to compute $\partial\mathcal{Z}/\partial K|_{(\sigma,K)}$ analytically. A full overview of these equations with derivations can be found in Appendix A.3.

The result of solving the adjoint equations is a differential that lets us evaluate the first-order change to a conjugate point σ implied by changing K, up to a proportionality factor:

$$\sigma[K + \delta K] \approx \sigma[K] + \delta\sigma[\delta K] = \sigma[K] + \delta F[\delta K]/(-\partial\mathcal{Z}/\partial s|_{(\sigma,K)}).$$

This differential, which we call δF, can be used to compute a search direction to increase σ in an optimization algorithm.

4.5.3 A Failed Attempt at Stability Recovery

The most obvious way of using δF is to parametrize K by densely sampling it, and assigning one parameter K_i per sample. With this finite parametrization, δF gives the gradient ∇F with respect to K_i, which can be used for "steepest ascent" on σ. The data flow of this algorithm is

$$K \mapsto (\gamma, \lambda) \mapsto (\mathcal{Z}, \sigma) \mapsto F.$$

Algorithm 4.1, which outlines this idea, makes initial progress in moving σ closer to l, but necessarily fails to cross the stability threshold. The reason is the bifurcation that occurs between unstable and stable equilibria at $\sigma = l$. At this point, the curve becomes "uncontrollable" through K, meaning that small changes to K may induce arbitrarily large changes to γ, and γ may not even be (locally) uniquely defined by K. The consequence is that Newton-type methods that compute γ from K diverge, and even if they are stabilized, any similarity of γ to the original curve is lost in the process.

Theorem 1 offers the tools to tackle the stability optimization problem with a more successful approach: It describes the exact requirements on γ under which the inverse map $\gamma \mapsto K$ exists. This map is well-behaved even through bifurcations, and we will use it to construct a working algorithm for stability recovery below.

4.5.4 The Stability Recovery Algorithm

The key to repairing the stability recovery algorithm is to avoid the "forward simulation" step, in which γ is computed from K. To do this, we exchange the primary variables of the optimization problem, and parametrize γ instead of K. We opt for a B-Spline representation of at least quartic order, and use the control point coordinates $q \in \mathbb{R}^{2m}$ as parameters to optimize, with m the number of control points. The data flow of the new algorithm is

$$q \mapsto \gamma \overset{!}{\mapsto} (K, \lambda) \mapsto (\mathcal{Z}, \sigma) \mapsto F.$$

The step marked by "!" is only well-defined if γ is an equilibrium curve, because otherwise, an admissible stiffness function K does not exist. This shows the significance of Theorem 1—it gives necessary and sufficient conditions that we can enforce computationally in order to retain the equilibrium property of γ.

In practice, this means enforcing conditions (1) and (2) of Theorem 1 throughout optimization. The collinearity of inflection points can be formulated as a non-linear equality constraint on q,

Algorithm 4.1: Failed Attempt at Stability Recovery

Input : equilibrium curve γ^0, stiffness K^0, Lagrange multipliers λ^0, step size h
Output : a stable equilibrium curve γ^n

$n \leftarrow 0$;
while γ^n *unstable* **do**
  Compute $\mathcal{Z}$, σ;
  Compute ∇F via adjoint method;
  $K^{n+1} \leftarrow K^n + h \cdot \nabla F$;
  Compute γ^{n+1}, λ^{n+1} based on K^{n+1}, with γ^n, λ^n as initial guess;
  $n \leftarrow n + 1$;
end

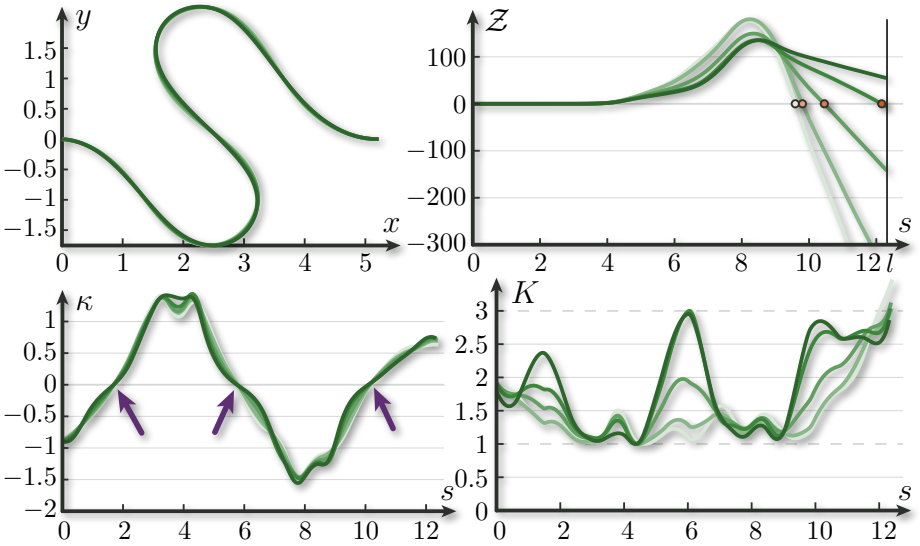
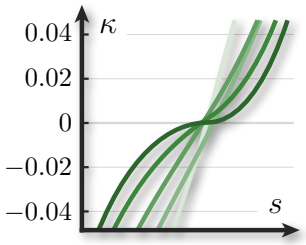


Figure 4.10: **Stability Optimization.** Iterations of stability optimization, from initial unstable state (light green) to stable state (dark green). Changes to γ *(top left)* are almost imperceptible, but its curvature *(bottom left)* changes visibly around inflection points (arrows). This causes stronger stiffness variations *(bottom right)* and pushes the conjugate point (orange) out of $(0, l]$ *(top right)*. Iterations 0, 8, 15, 23, 30 shown.

and is enforced via an underdetermined Newton procedure after every update $q \leftarrow q + \Delta q$. This step is also used to enforce other constraints, such as keeping endpoints and tangents fixed, and fixing the arc length of the curve. We describe this procedure in Appendix A.4.



At this stage, the algorithm converges for some examples, but fails for others. The reason for failure is that the optimization often drives γ towards a state in which the secant-boundedness of κ is violated, i.e., with $\kappa'(s_0) \approx 0$ at an inflection point s_0, as seen in the inset. The easiest way to remedy this, and safeguard against other violations of (1) and (2), is to add a constraint that bounds $R[K] = \frac{\sup K}{\inf K}$ from above. The upper bound $R_{\max}$ can be chosen to reflect the limitations of the fabrication method, or other requirements on the geometry of the rod. This is a single non-linear inequality constraint, and it can be enforced by linearizing the constraint in q and projecting Δq onto the tangent space of the constraint boundary, if it violates the constraint to first order. We have found that it is not necessary to iteratively project q back onto the (non-linear) constraint boundary, because violations of $R \leq R_{\max}$ due to the linearization are typically small and temporary if the step size is kept small enough. Algorithm 4.2 summarizes all steps of computation.

Example Fig. 4.10 shows iterations of optimizing the stability of a quartic spline curve with three inflections, where $R_{\max} = 3$. Remarkably, the curve itself changes very little during optimization, but the stiffness profile implied by the curve changes drastically. This is possible because curvature changes close to inflection points are amplified by the computation $K = \frac{a + \langle b, \gamma \rangle}{\kappa}$.

The first two photos in Fig. 4.11 show a physical model of the unstable elastic curve, which can be kept in unstable equilibrium with a small amount of force, but otherwise snaps into one of the stable equilibria. The last two photos show a model of the stabilized curve, whose

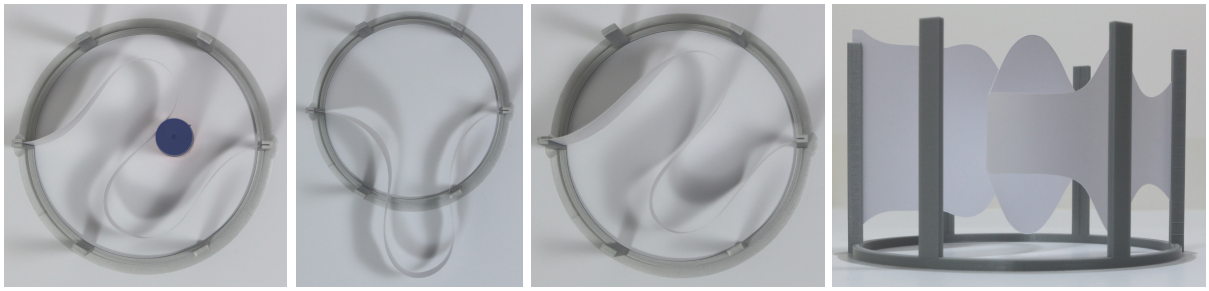


Figure 4.11: **Stabilized *S*-curve.** *From left to right:* Initial curve is kept in its unstable equilibrium state by a little friction; initial curve snaps into stable equilibrium once friction is removed; optimized curve is stable in *S*-shape; side view of stable *S*-shape.

equilibrium shape is almost identical to the unstable one. Note that the model does not touch the ground, so no frictional force is provided.

Remarks There are two technical details that we have brushed over so far. First, instead of using $R[K] = \frac{\sup K}{\inf K}$ directly, we replace it by the differentiable approximation $R[K] \approx \|K\|_p \cdot \|1/K\|_p$, with $\|f\|_p = (\int |f|^p)^{1/p}$ and p large; $p > 20$ worked well in our examples.

Second, we did not carefully discuss differentiability of $\gamma \mapsto (K, \lambda)$. This map is only defined on the manifold $\mathcal{M} \subset \mathbb{R}^{2m}$ corresponding to control point coordinates q such that γ has collinear inflections. Thus, the domain of the differential is the tangent space $T_q\mathcal{M}$, and not all of $\mathbb{R}^{2m}$. In our implementation, we account for this by computing an explicit basis of $T_q\mathcal{M}$ after enforcing constraints, and performing all subsequent calculations within this subspace.

A current limitation of the stabilization algorithm is that it neglects gravity because it uses Theorem 1 to formulate constraints. Heuristically, we can account for gravity by enforcing Eq. 4.5 after stabilization while changing K minimally, but this gives no formal stability guarantee.

Algorithm 4.2: Stability Recovery

Input : equilibrium curve γ^0, step size h, stiffness ratio bound $R_{\max}$
Output : a stable equilibrium curve γ^n

$n \leftarrow 0$;
while γ^n *unstable* **do**
  Compute $a \in \mathbb{R}, b \in \mathbb{R}^2$ from inflection line;
  $K \leftarrow \frac{a+\langle b, \gamma^n \rangle}{\kappa}$; $R \leftarrow \frac{\sup K}{\inf K}$;
  Compute $\mathcal{Z}$, σ;
  Compute $\delta F / \delta K$ via adjoint method;
  Compute $\mathrm{d}K/\mathrm{d}q$ and $\mathrm{d}R/\mathrm{d}K$ via chain rule;
  $\nabla F \leftarrow \delta F/\delta K \cdot \mathrm{d}K/\mathrm{d}q$; $\nabla R \leftarrow \mathrm{d}R/\mathrm{d}K \cdot \mathrm{d}K/\mathrm{d}q$;
  $\Delta q \leftarrow h \cdot \nabla F$;
  // Project Δq so $R \leq R_{\max}$ is satisfied to first order
  $\Delta q^{\text{eff}} \leftarrow \textsc{ProjectStep}(q, \Delta q, R, \nabla R, R_{\max})$;
  $q \leftarrow q + \Delta q^{\text{eff}}$;
  $q \leftarrow \textsc{EnforceConstraints}(q)$; // Make inflections collinear, enforce b.c.
  $\gamma^{n+1} \leftarrow \textsc{BSpline}(q)$; $n \leftarrow n + 1$;
end

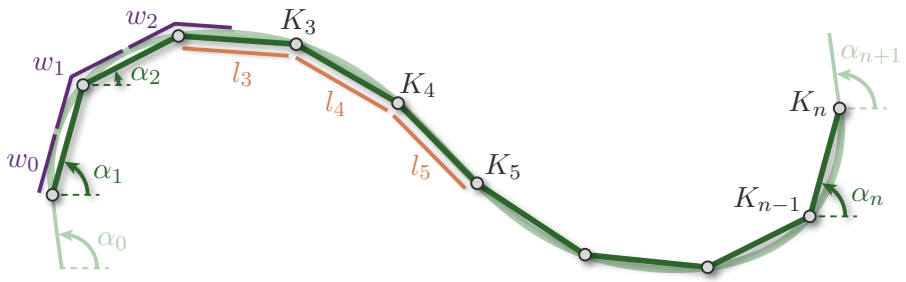


Figure 4.12: **Discretization.** Quantities used in the discretization of the bending energy: edge and boundary angles (green), edge weights (orange), nodal stiffnesses (gray), nodal weights (purple).

4.6 Implementation

Here, we give details about the discretization and form-finding algorithm that are part of our design system.

4.6.1 Discretization

Our discretization of plane elastic rods is similar to that of Bergou et al. [BWR+08], and we distinguish between nodal quantities and edge quantities. This discretization is not needed for implementing the results from Sections 4.3 and 4.4, but we use it to discretize the stability criterion and its adjoint (Section 4.5), as well as for numerical testing with forward simulation.

A curve is represented as a polygonal chain (Fig. 4.12) with edge lengths l_i, which function as weights for edge quantities. Nodal quantities have as weights the Voronoi area w_i of a node, i.e., half the sum of incident edge lengths. The bending energy and endpoint constraints of a rod are discretized as

$$W = \frac{1}{2}\sum_{i=0}^{n} K_i \frac{(\alpha_{i+1}-\alpha_i)^2}{w_i} \quad \text{and} \quad \sum_{i=1}^{n} l_i \begin{pmatrix}\cos\alpha_i \\ \sin\alpha_i\end{pmatrix} = \gamma_l,$$

where $K_0, \ldots, K_n$ are nodal stiffnesses; $\alpha_1, \ldots, \alpha_n$ are edge angles; and α_0 and α_{n+1} are angle boundary values. The solution ζ to Eq. 4.7 is an edge quantity, with one fictitious edge added on either side of the curve. This gives samples $\zeta_0, \ldots, \zeta_{n+1}$, with $\zeta_0 = 0$ and $\zeta_1 = w_0$ by the initial conditions, and

$$-\frac{1}{l_i}\left(K_i\frac{\zeta_{i+1}-\zeta_i}{w_i} - K_{i-1}\frac{\zeta_i-\zeta_{i-1}}{w_{i-1}}\right) - \langle\lambda, \gamma_i'\rangle\zeta_i = 0,$$

with $\gamma_i' = (\cos\alpha_i, \sin\alpha_i)$, which can be solved for ζ_{i+1}. The same edge discretization is used for η_i. Finally, M_i and N_{ij} are nodal quantities, which are computed by summing over edges in the same manner as for the endpoint constraints. The adjoint quantities can be discretized in the same way as the primal quantities. To compute $\delta F[\delta K]$, the derivatives $\alpha', \bar{\alpha}', \zeta', \bar{\zeta}'$, etc. are computed as nodal quantities, and initial values like $\zeta_0' = 1$ can be recovered exactly from the fictitious border edges that were added earlier.

4.6.2 Implementation of the Design System

The algorithms for computing a stiffness profile with or without gravity can be implemented by sampling an input curve and setting up the corresponding linear programs from Eqs. 4.4 and 4.6. We solve these programs to optimality using the Gurobi library.

Our design system offers different modes of user interaction. On the one hand, the user can directly edit a design by manipulating the control points of spline curves or surfaces, and modifying planes that are intersected with input surfaces to yield curves. The program generates a preview of the beam geometry necessary to realize these curves in a split second (see Fig. 4.16), which allows for a fast and interactive workflow.

The user can also give more control to the application by running a fabricability optimization routine that improves the design automatically. This is especially useful if the optimal stiffness ratio $R = \frac{\sup K}{\inf K}$ of a curve is too high for the chosen fabrication method, and the user cannot decrease it further with manual edits. At the core of this optimization method is a local/global approach.

The local step computes the derivative of R with respect to coordinates of the spline control points, and modifies them using gradient descent with line search. The parameters a and b are modified as little as possible in the process. E.g., if there is a single inflection point, only b is modified so L still intersects the inflection point after the control points have been updated. These minimal changes to a and b are accounted for when computing the gradient.

The global step solves the linear program in Eq. 4.4 to regain the global optima for a and b, while keeping the control points fixed. This step is only necessary if there are fewer than two inflection points, because otherwise a and b are defined up to positive multiples by the inflection line.

Neither step can increase R, given a small enough step size. For inflectional curves, we call EnforceConstraints after the local step. Optimization stops once a user-defined target stiffness ratio has been reached, or after a fixed number of iterations. We account for gravity in a post-process, by solving for K once more with the linear program from Eq. 4.6 after the local/global algorithm terminates.

Naturally, the user can mix these modes of interaction freely, in order to converge to the best possible design. Once a design is finalized, the system outputs CAD files that include the outlines of all beams in their flat configuration, and the placement and orientation of fixtures for all beam endpoints.

4.7 Fabrication & Validation

Next, we present ways of manufacturing elastic strips that exhibit the stiffness profiles we have computed in previous sections. We show two methods that let us control the stiffness and width of a strip independently, by using either perforation or layering. To quantify the accuracy of deformed shapes, we 3d scanned physical models and compared them with the intended design. We also present numerical tests to assess the robustness of fabrication using elastic strips. Extensions that allow the use of nonlinearly elastic materials and account for mild plasticity effects are discussed in Appendix A.5.

4.7.1 Fabrication

The most straight-forward way of producing an elastic strip is to cut it from a sheet with constant thickness h, such that the width $w(s)$ of the strip is directly proportional to the stiffness $K(s)$. This achieves the desired behavior because $K \sim wh^3$ for elastic rods with a rectangular cross section.

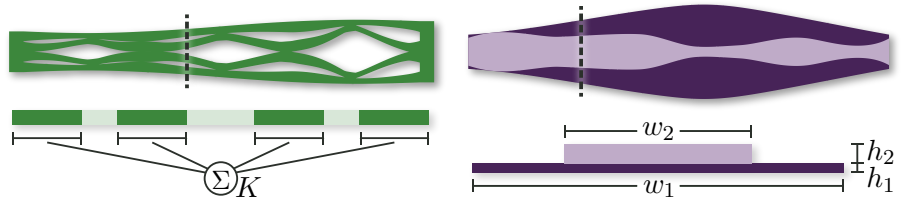


Figure 4.13: **Perforated and Composite Strips.** *Top:* Examples of perforated *(left)* and composite *(right)* strip with section lines. *Bottom:* Cross sections.

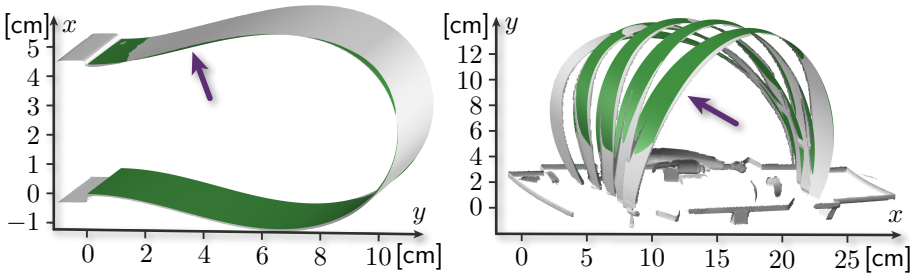


Figure 4.14: **3d Scans.** Scans of physical models (gray) superimposed with simulated equilibrium shape (green). Arrows show maximal displacement error. *Left:* Cardboard model of arch with two inflections (max err. 1.7 mm). *Right:* Five interlocking arches of pavilion example (max err. 2.5 mm).

The disadvantage of this method is that we relinquish direct control over the appearance of the strip, because all degrees of freedom go into realizing the stiffness. In particular, this makes it impossible to close gaps between adjacent strips in a model, because we cannot control the width independently of the stiffness. We found two ways to circumvent this limitation, shown in Fig. 4.13.

Perforated Strips To prescribe the width and stiffness independently, we can remove material from a strip with width $w(s)$ by adding holes in order to achieve an *effective stiffness* $K(s)$. Our design removes diamond-shaped regions of material to reveal a network of smaller-scale strips, such that the total width integrated over a cross section is proportional to K, see Fig. 4.13 (left). We chose this type of perforation so the smaller-scale strips are approximately aligned with the direction of the main strip, in order to ensure uniform bending.

Composite Strips Another way of decoupling w and K is to use a composite of two strips that are rigidly glued, see Fig. 4.13 (right). We use a broader strip (purple) with thickness h_1 and width $w_1(s)$ to determine the outer shape of the strip, and a narrower strip (lilac) with thickness h_2 and width $w_2(s)$ to add stiffness control. The stiffness of the composite strip can be approximated by[3]

$$cK = w_2(h_1 + h_2)^3 + (w_1 - w_2)h_1^3 = P_1 w_1 + P_2 w_2,$$

with $P_1 = h_1^3$, $P_2 = (h_1 + h_2)^3 - h_1^3$, and c a scaling constant. We want to prescribe K and w_1, and compute w_2 subject to $0 \leq w_2 \leq w_1$ to satisfy this equation with some $c > 0$. The

[3]In this formula, we neglect the small offset between the overhang of the broad strip and the neutral line of the cross section. Instead, we assume that all mass is symmetrically distributed around the neutral line.

bounds on w_2 lead to constraints on c,

$$c_{\min}(s) := \frac{P_1 w_2(s)}{K(s)} \leq c \leq \frac{(P_1 + P_2) w_2(s)}{K(s)} =: c_{\max}(s),$$

for all $s \in [0, l]$. These constraints can be satisfied simultaneously if and only if $\max c_{\min} \leq \min c_{\max}$. If this is the case, we use $c := \min c_{\max}$, which makes the narrow strip as wide as possible without protruding beyond the broad strip, as shown in Fig. 4.13 (right). Should the problem be infeasible, one can either change the design or reduce the thickness of the broad strip relative to that of the narrow strip. This works because the problem always becomes feasible as $h_1/h_2 \to 0$. All of our results that use composite strips were realized with $h_1/h_2 = 1/2$.[4]

4.7.2 Validation

3d Scans. To better quantify the predictive capabilities of our model, we used a dual-camera light scanner to capture the geometry of two physical models in their deformed state. The scanned models were then registered to the 3d models that served as input for computing the stiffness profiles. Fig. 4.14 shows a superimposition of the scanned and input models.

The first scan is of an arch with two inflections and a total length of 25 cm, yielding a maximum displacement error of 1.7 mm; this is about 1% of the model diagonal. The second scan shows a section of the pavilion example, composed of five interlocking elastic strips, each approximately 34 cm in length. Here, the maximum displacement error is 2.5 mm, also about 1% of the model diagonal.

Numerical Testing. In addition to conducting 3d scans, we performed a series of numerical tests that simulate the effects of material parameter uncertainty and fabrication error. This helps analyze the robustness of the fabrication process with elastic strips. Fig. 4.15 shows a few representative equilibrium curves, taken from the examples we show in Section 4.8. They include curves with zero to three inflection points, and one curve with a high turning number. In particular, we show the consequences of the following defects (rows 3–6):

- The ratio between minimal and maximal stiffness is lower (higher) than assumed.
- The gravity parameter $\frac{\rho}{Eh^2}$ is lower (higher) than assumed.
- The enforced boundary angles are defective.
- The highest-curvature regions deform plastically.

The exact impact of these errors will depend on the equilibrium curve in question, but we found curves with high total variation of turning angle, $\mathrm{V}(\alpha) = \int_0^l |\kappa|$, to be affected the most. On the other hand, inflection points close to the endpoints of a curve (e.g., column 4) seem to have a stabilizing effect.

4.8 Results

We combine the design methods from Sections 4.3–4.6 and the fabrication techniques from Section 4.7 to manufacture physical models that demonstrate applications in architecture,

[4] The gravity potential of a composite strip is linear in the unknown, w_2, so the model from Section 4.4 can easily be adapted to solve for w_2 of a composite strip with w_1 given.

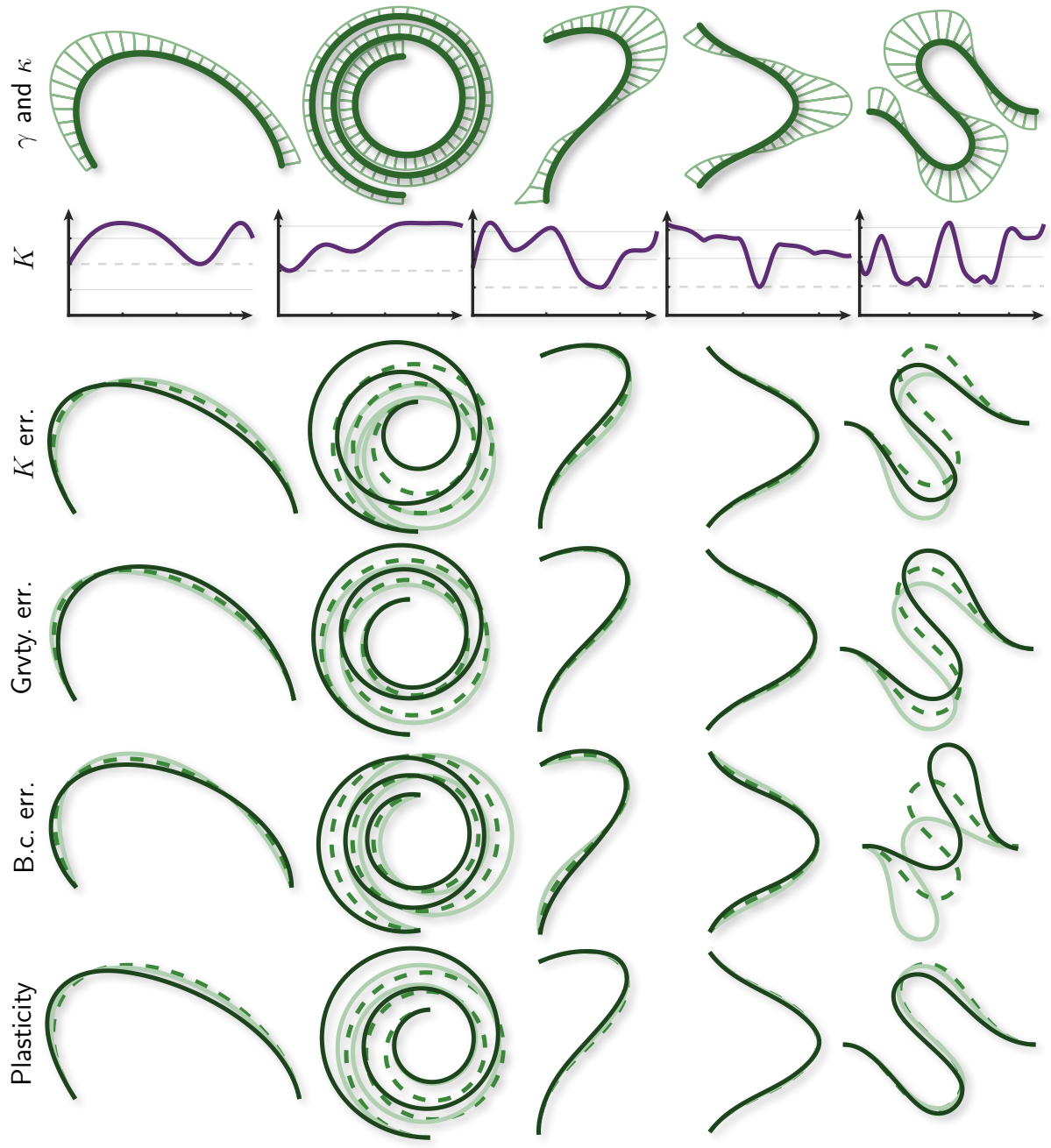


Figure 4.15: **Robustness Study.** We show the effect of fabrication errors and parameter uncertainties on elastic curves on five representative examples. *Rows 3–6*: original curve (dashed) and curve affected by errors. *Top to bottom*: Curve and curvature normals; stiffness profile; stiffness range smaller (light green) or larger (dark green) than intended by 50%; gravity parameter underestimated (light) or overestimated (dark) by 50%; boundary angles off by 10°; plasticity affecting top 25% (light) or top 50% (dark) of curvature.

Table 4.1: **List of Results.** Summary of all models. Design: Curves based on 3d models, on analytical expression, or modeled directly by hand. #Strips: Total (Individually optimized). #SpC: Number of samples per curve. Size: length x width x height. Timings for stiffness computation, fabricability optimization, geometry processing, and sum of all three.

Name	Material	Design	Grvty.	Fabrication	#Strips	#SpC	Size [cm]	Stfns.	Fbrcblty.	Gmtry.	Total
Shell	Cardboard	Model	No	Perforated	60 (10)	200	50x49x10	0.05 s	1.39 s	0.57 s	2.01 s
Flower Pot	Cardboard	Model	Yes	Composite	16 (16)	500	30x27x17	1.26 s	13.3 s	1.46 s	16.0 s
Horse	Cardboard	Model	No	Composite	40 (40)	200	56x14x39	0.15 s	2.84 s	11.6 s	14.6 s
Pavilion	Cardboard	Analytical	Yes	Emergent	31 (31)	200	91x23x14	0.30 s	-	3.93 s	4.23 s
Lantern	POM	Direct	No	Emergent	20 (2)	240	28x28x26	0.003 s	0.47 s	-	0.47 s
Lamp	Plywood	Model	No	Perforated	32 (32)	200	93x66x17	0.11 s	14.0 s	0.12 s	14.2 s
Vase	Cardboard	Analytical	No	Emergent	20 (1)	1.3k	26x26x14	-	-	-	0.01 s
Carpet	-	Model	Yes	-	15 (15)	400	46x29x12	0.89 s	2.47 s	1.76 s	5.12 s
Façade	-	Direct	No	-	8 (8)	300	55x12x43	0.17 s	7.36 s	1.33 s	8.86 s

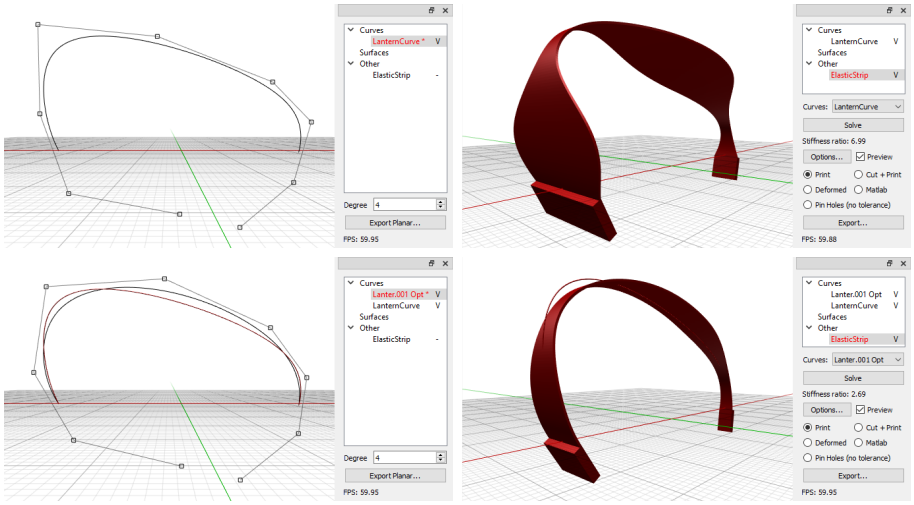


Figure 4.16: **Lantern Design Session.** *Top:* B-Spline curve modeled by the user (left) and optimal stiffness profile computed by Eq. 4.4 (right). *Bottom:* Result of automatic fabricability optimization to guarantee $R \leq 3$.

model building, and interior design. Our stiffness construction algorithm reacts to user edits within a fraction of a second, which allows for fast iteration on the design of these models, and a quick evaluation of ideas. We also use the fabricability optimization routine of our design system (see Section 4.6.2) in order to suggest trade-offs between the original concept and ease of fabrication. For a detailed summary of all results, along with material, size, complexity, and computation times, consult Table 4.1.

We showcase a variety of materials such as cardboard, paper, polyacetal, and plywood, as well as different design processes such as approximation of existing 3d models, and direct specification using mathematical expressions. Some models use perforated or composite strips for maximal control, while others benefit from the naturally emergent shapes dictated by the stiffness distribution. Many of the models are between 50 and 100 cm in length, which would make it cost-prohibitive to 3d print all beams in their curved state, or to manufacture molds for all of them. Photographs of the physical models are shown in Figs. 4.1 (left) and 4.24, and renderings of two additional examples in Figs. 4.1 (right) and 4.23.

4.8.1 Emergent Strip Designs

We designed two objects that take their elastic strip geometry directly from the stiffness profiles computed with linear programming.

Lantern The first is a lantern encased by twenty strips lasercut from a POM sheet with a thickness of 0.5 mm. The shape of the elastic strips was drafted in our software design tool using B-Spline curves, as seen in Fig. 4.16. The tool gives immediate visual feedback about the geometry of the strip resulting from the current design. The user can then either manually adapt the design to achieve better fabricability and appearance, or use the automatic optimization routine to decrease the max-to-min stiffness ratio of the strip. For this example, the optimization terminates under 1 second, giving real-time feedback and enabling an iterative design loop.

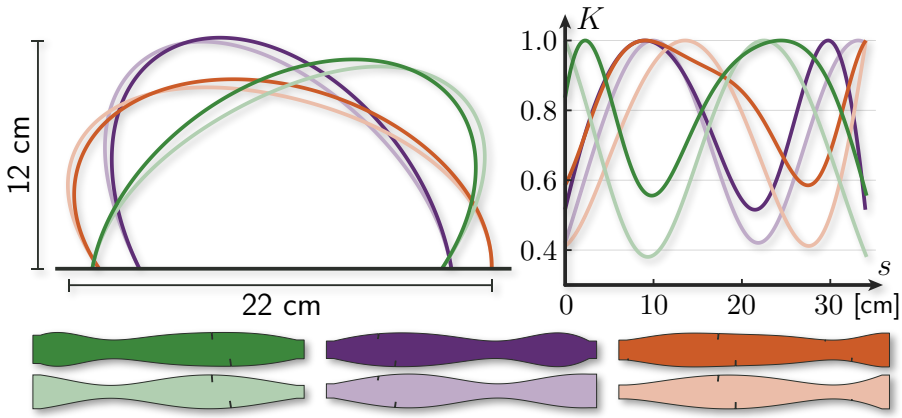


Figure 4.17: **Pavilion.** *Top left:* Three arches from the pavilion model (dark); poor shape reproduction if gravity is neglected during optimization (light). *Top right:* Stiffness profiles computed with gravity (dark) and without gravity (light). *Bottom:* Final strip geometry with cuts for interlocking.

Pavilion The second object is a pavilion composed of 31 interlocking cardboard strips that form an archway about one meter long, and 15 cm high. Every strip is designed as a segment of an ellipse, rotated around its center point, which gives the model a corkscrew-shaped appearance. The design uses 200 gsm cardboard, which is flexible enough for gravity to play a significant role, so we used the model from Section 4.4 to compute the width profile of every strip. Fig. 4.17 shows the effect of gravity, along with the resulting stiffness and width profiles.

The strips are spaced closely enough so neighboring strips intersect by a few millimeters each. These intersections are found computationally and show up as slotted cuts in the final model, to allow the strips to interlock. Note that little to no force is transmitted through these cuts, because every strip is in equilibrium even without neighboring strips as support. Fig. 4.14 validates the physical shape by 3d scanning.

4.8.2 Composite Strip Designs

If the goal is to cover a closed 3d model with elastic strips, it is useful to have explicit control over the width profile of every strip, so gaps between strips can be minimized. We can achieve this by using composite strips, as discussed in Section 4.7, and demonstrate the technique on two 3d models.

Horse This object is based on a pre-existing 3d model of a horse[5] that was not modified by the authors in any way prior to approximating it with equilibrium curves. To design the strip model, we specify a family of twenty planes spanning the body, neck, and head of the model, and compute the intersection curves, as shown in Fig 4.18. All remaining steps are automatic, except for choosing parameters such as spline curve degree and step sizes.

Each intersection curve is split at the spine, yielding a total of fourty sampled curves, which are then smoothed to remove high frequencies and approximated by quartic spline curves. Curves that do not have the equilibrium curve property initially are post-processed by removing spurious inflection points. Then, we run the auxiliary routine to optimize fabricability (see Section 4.6.2) until a value of $R \leq 2$ is reached for each curve. As Fig. 4.18 shows by example,

[5] https://free3d.com/3d-model/palomino-horse-walking-v1–643031.html

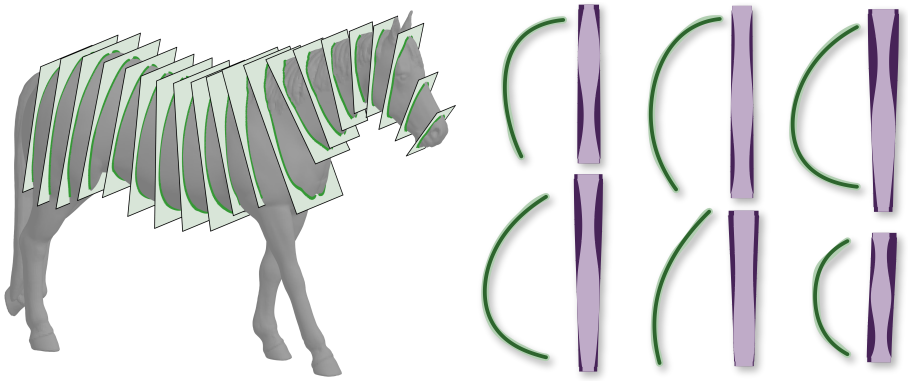


Figure 4.18: **Horse.** *Left:* 3d model with planes and intersection curves. *Right:* A selection of input curves (light green) and optimized equilibrium curves satisfying $R \leq 2$ (dark green); composite strips realizing these curves (purple).

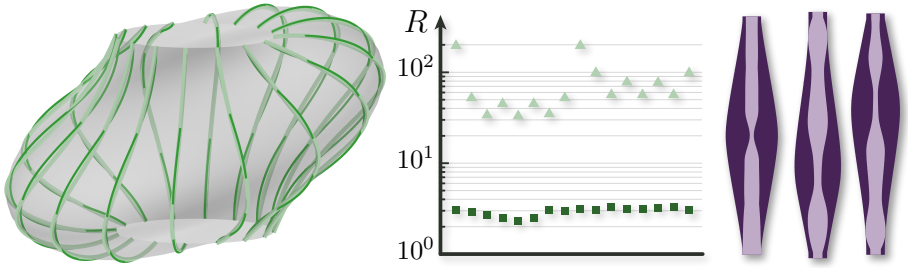


Figure 4.19: **Flower Pot.** *Left:* NURBS model with input curves (light green) and optimized curves (dark green). *Center:* Stiffness ratio objective R of all curves before optimization (light) and after optimization (dark). *Right:* Three composite strips used in the physical model.

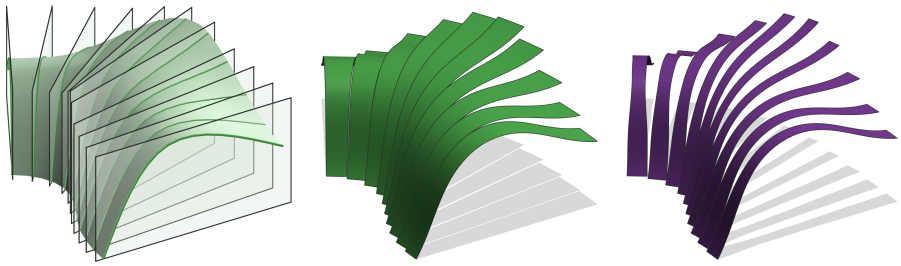


Figure 4.20: **Shell.** *Left:* 3d model partitioned into ten sections. *Center:* Coverage of each elastic strip after fabricability optimization, using composite or perforated strips. *Right:* Sparse coverage if width is proportional to stiffness.

most curves only change slightly as a result. The final step is to solve for w_2 of the composite strip as described in Section 4.7.1.

The physical model consists of 3d printed parts for the legs and tail of the horse, as well as a slender frame structure to hold the elastic strips. The composite strips are cut from 200 gsm cardboard and 100 gsm paper using a Cricut cutting machine, and then glued in their flat state. After insertion in the frame, they form the body, neck, and head of the horse. Some strips are tilted relative to the direction of gravity. This causes out-of-plane forces, but their effect is negligible on this scale, so planarity of deformations is retained.

Flower Pot We designed this model by intersecting an asymmetric NURBS surface with radially arranged planes, which gives a total of 16 curves that we want to approximate as

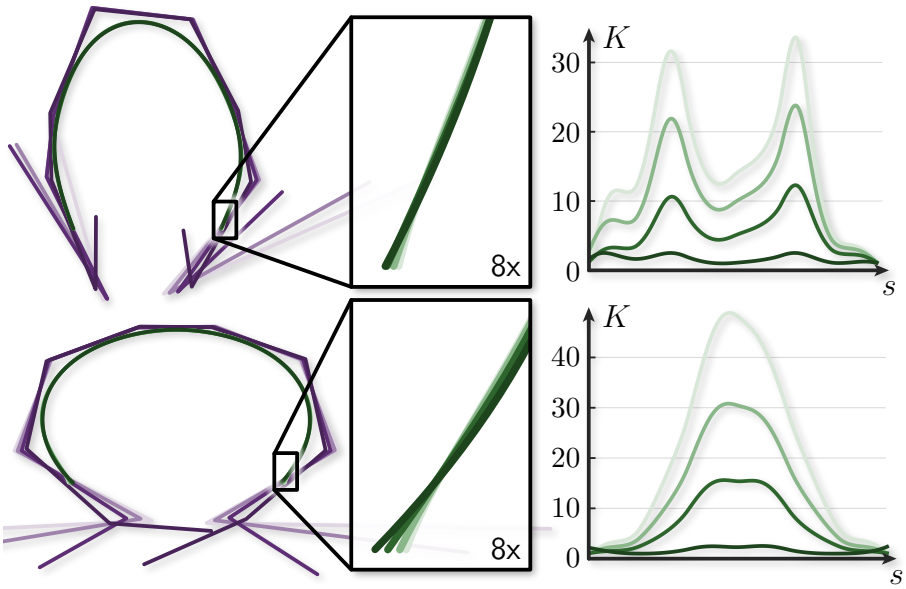


Figure 4.21: **Lamp Optimization.** *Left:* Evolution of equilibrium curve (green) and its control polyline (purple) during fabricability optimization, from initial state (light) to final state (dark). *Right:* Evolution of stiffness profile K, normalized to $\min K = 1$.

closely as possible with elastic strips (see Fig. 4.19). All curves initially satisfy the conditions of Theorem 1, but the optimal values of R lie between 30 and 110, making it impractical for fabrication.

We opt to run automatic fabricability optimization to guarantee $R \leq 3.5$ for all 16 strips, which takes 13 s in total. The curves change slightly in the process, as shown in Fig. 4.19, but the overall shape of the object is preserved. Finally, we use the technique from Section 4.4.3 to arrive at a smooth stiffness profile, and compute the shapes of composite strips as discussed in Section 4.7.1.

4.8.3 Perforated Strip Designs

Another way of decoupling stiffness and strip width is to use perforation, which gives the physical object a more stylized appearance. We demonstrate this technique by manufacturing a miniature of an architectural shell from cardboard, and a large ceiling lamp made from plywood.

Shell This model is based on a curved shell with three-fold mirror symmetry, initially given as a quad mesh. Fig. 4.20 (left) shows one sixth of the shell, along with a partition into ten slender sections. Each section grows wider towards the far end, which would cause large gaps between strips if the stiffness profile was used to directly determine strip width. To achieve better coverage (center), and an even distribution of material across the shell, we use the perforation technique described in Section 4.7.1.

The physical model shown in Fig. 4.24 (row 2) has a uniform texture with no large gaps and contains regions of both negative and positive apparent curvature. In total, sixty strips of 200 gsm cardboard with diamond-shaped cutouts are used to form the curved surface.

Lamp Our largest model, with a footprint of 93 x 66 cm and a height of 17 cm, uses a 3d printed base with a thickness of 7 mm, and 32 elastic strips that have been lasercut from 0.8

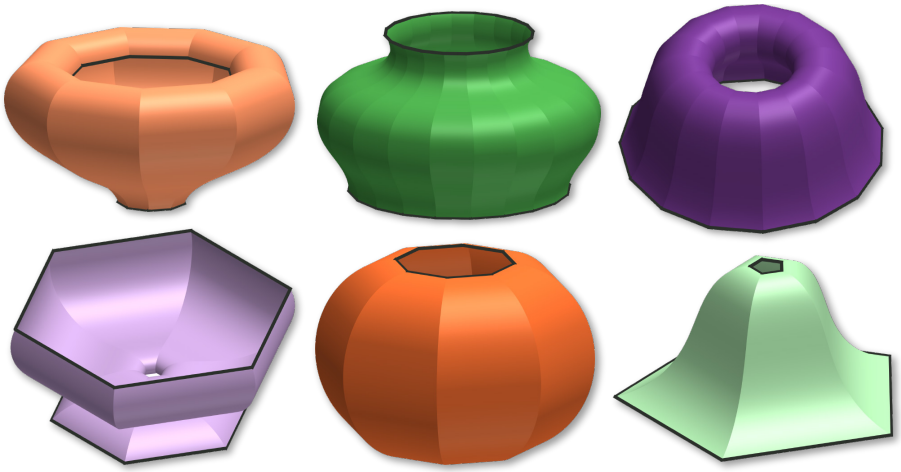


Figure 4.22: **Surfaces of Revolution.** Examples of surfaces of revolution that can be tessellated without gaps. Lines mark kinematic boundary conditions.

mm plywood. Furthermore, the base is clad in a plywood veneer, so the lamp appears to be entirely done in woodwork. Without the use of active bending, a model like this would be extremely costly to make, requiring individual molds for all curved elements.

The design takes an existing shell model [GGP+20], which was not modified by the present authors, and approximates planar sections with equilibrium curves. As shown in Fig. 4.21, the stiffness profiles of these curves can be significantly improved by tiny changes near the endpoints. Our optimization algorithm does this automatically, in less than 0.5 s per curve.

4.8.4 Surfaces of Revolution

Elastic strips with constant thickness can be used to tessellate a family of surfaces of revolution without gaps. We can characterize this family with a small modification to Eq. 4.3. Note that the tessellation constraint requires the width $w(s)$ of each strip to be directly proportional to the distance from the axis of revolution (cf. Fig. 4.22). If we identify the axis of revolution with e_2, and a direction orthogonal to it with e_1, this constraint reads $\langle e_1, \gamma(s) \rangle \sim w(s)$, where w is proportional to K. Plugging this into Eq. 4.3 yields

$$\langle e_1, \gamma \rangle \kappa = a + \langle b, \gamma \rangle,$$

where the proportionality constants have been absorbed into the right-hand side. We can use this equation to generate all surfaces of revolution with the tessellation property by picking a, b, and initial conditions $\gamma(0)$ and $\alpha(0)$. Then, we integrate through the equation to solve for a meridian of the surface.

This application was inspired by Liu et al. [LDV20], who show a construction for surfaces of revolution with at most one inflection point. As the examples in Fig. 4.22 and our physical model in Fig. 4.24 show, our construction also supports more than one inflection, assuming kinematic boundary conditions. The subfamily with symmetric b.c. on one end is described by restricting b to multiples of e_2.

4.8.5 Applications in Lighting Design

Figs. 4.1 (right) and 4.23 show two more applications of active bending in the form of renderings. The first is a flowing pavilion design ("Carpet") realized with 15 inflectional curves

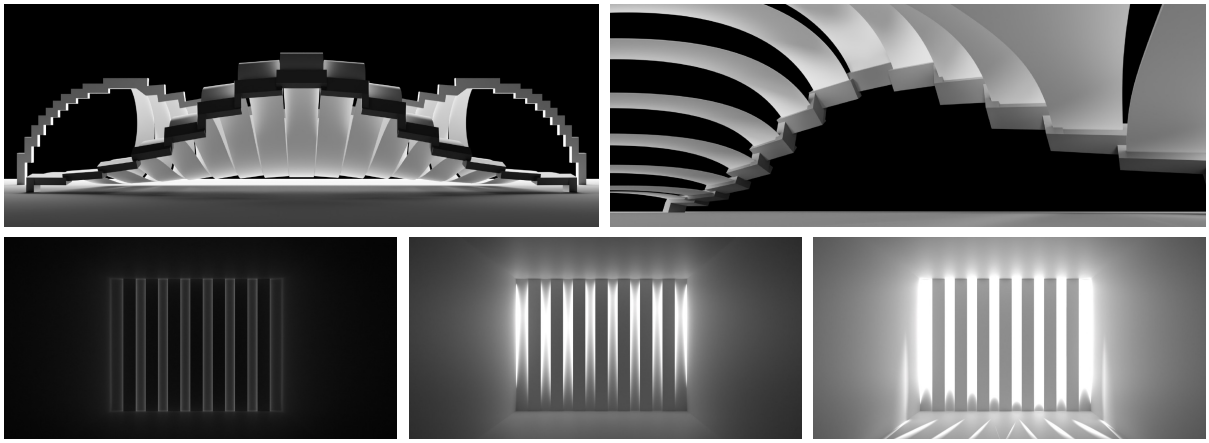


Figure 4.23: **Renderings of Carpet and Façade.** *Top:* Carpet viewed from the front, and from the inside out. *Bottom:* Façade closed, partly open, and fully open.

whose lowest points hover just above ground. The gap in-between allows indirect light to enter through the back and flood the space underneath the structure.

We further explore the role of active bending in lighting design with a kinetic façade inspired by the work of Knippers et al. [KSOJ12]. Elastic beams are placed vertically to form a façade that can be actuated by compression, in order to control the amount of indirect light entering through. We used our design system to determine the deformed shape of the beams, to allow more light to enter through the top of the mechanism, as seen in Fig. 4.23 (bottom center).

4.9 Discussion

In this work, we characterized the design space of plane elastic curves. Different spatial arrangements of these curves give rise to a variety of appealing forms, but we have yet to explore the possibilities offered by connecting elastic strips with joints, allowing them to undergo torsion, and accounting for the effect of creep over time. Understanding the complex design space offered by these mechanisms in geometric terms may offer new ways to support designers and enable flexible workflows with quick feedback loops. We believe that our approach to characterizing plane elastic curves geometrically can be generalized to the case of several jointed curves and non-planar deformations by considering appropriate variational problems and constraints.

Another useful extension would be to integrate standard beams, bending-active beams, and external forces in a single system. Possibly, our geometric characterization of bending-active beams can be combined with graphical statics to promote an intuitive understanding of these mixed constructions. Furthermore, the present work does not consider the assembly process of bending-active structures. Deployment with light-weight mechanisms that respect stress bounds is an active area of research, and an essential part in the scaling up of designs.

In terms of application, we have only scratched the surface of kinetic structure design, in which actuation is used to alternate between different shapes. This idea offers an interesting challenge, because it is necessary to optimize structural elements for several deformed shapes at once, and to preserve the functionality of a design. A related problem is that of designing multi-stable elastic structures, which need only be actuated during a shape switch, but stay in each target shape without external force. We hope that our idea for stability optimization can contribute to this area of research.

Figure 4.24: **Photographs of Physical Models.** *Top to bottom, left to right:* Pavilion, Horse, Shell, Vase, Lantern, Flower Pot, Lamp.

CHAPTER 5

The Design Space of Kirchhoff Rods

5.1 Introduction

Rod-like elements are ubiquitous in natural and human-made environments, and form the building blocks of the world around us: from beams used in the construction of buildings, machines, and tools, to electrical cables, clothing fibers, tree branches, human hair, muscle tissue, and DNA strands. This omnipresence has led to a long and rich scientific history concerned with developing the tools to predict how an elastic rod of a given length and shape will bend and twist under different loads and constraints.

The history of investigating the inverse problem—to determine the initial shape of a rod whose equilibrium state is known—is considerably younger but no less ambitious: It lays the theoretical foundation for designing rods with a programmable deformed shape that is hard-coded into its geometry.

Motivation Most rods, beams, and ribbons used in construction and design are straight in their undeformed state and acquire curvature only through the process of bending and twisting. Straight rods are preferred because they can be manufactured more cheaply and with less material waste, and packed more easily for transport. To enable the inverse design

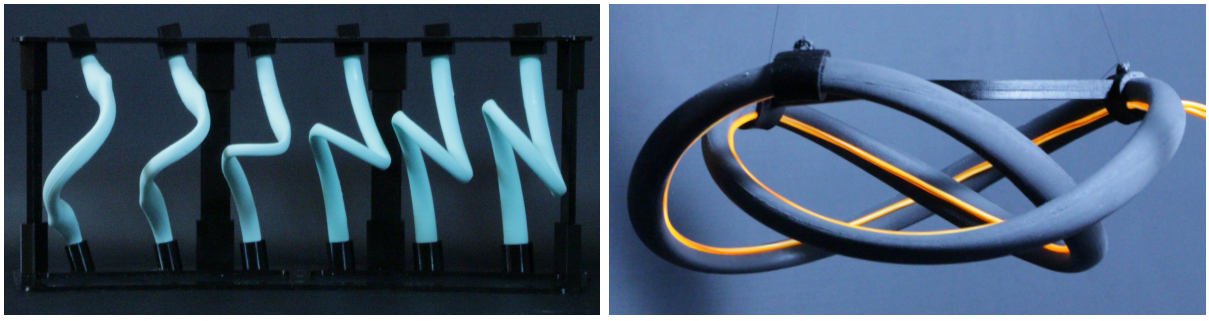


Figure 5.1: **Photographs of Physical Prototypes.** Our computational design algorithm takes as input a curve in $\mathbb{R}^3$ and computes the geometry of a straight elastic rod that will deform to match the input curve once installed in a support structure. *Left:* Six silicone rods that match six input curves subject to elastic forces and gravity. *Right:* Free-form light sculpture made from three straight silicone rods and electroluminescent wire, attached to a 3d-printed fixture.

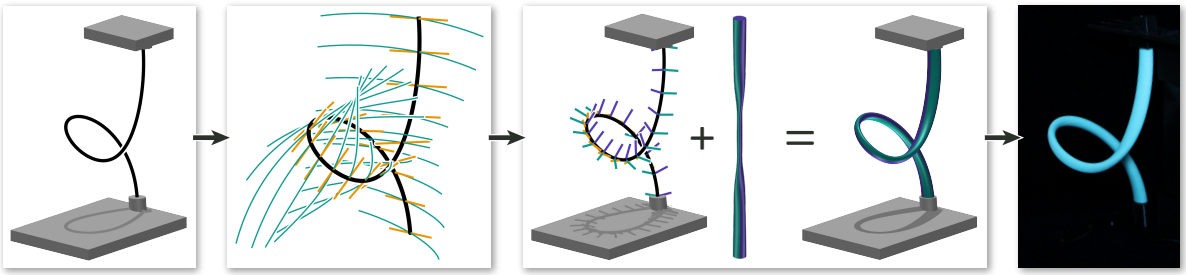


Figure 5.2: **Algorithm Overview.** *Left:* The input to our computational design algorithm is a curve in $\mathbb{R}^3$, which we want to convert to a deformed elastic rod. *Center-left:* We show that this problem has a solution if and only if there exists a helical motion (green) that is nowhere orthogonal to the binormal lines (yellow) of the curve. *Center-right:* Based on the parameters of the helical motion, we compute a moving frame along the curve and the geometry of a rod that solve the Kirchhoff rod equilibrium equation. *Right:* Photograph of a silicone rod that deforms according to the computed solution in the real world.

of straight rods with a programmable curved equilibrium shape, it is useful to know exactly which equilibrium shapes are physically realizable in the first place.

We solved this problem for the special case of planar deformations in Chapter 4, leading to a concise description of all plane curves that appear as static equilibria of straight elastic rods with spatially-varying thickness. This result also gave rise to a computational design algorithm, which solves for the thickness distribution necessary to achieve a given curved shape, and has applications in architecture and design.

A limitation of this approach is the restriction to planar shapes, and the inability to reproduce shapes that are curved in three dimensions. In this chapter, we show that a geometric characterization can also be attained for the case of fully three-dimensional bending and twisting of straight elastic rods with spatially-varying cross sections. This allows the computational design of structures such as the ones shown in Fig. 5.1, in which the curvature of the rods is induced purely by the elastic response of the material and the cross sections computed by our algorithm.

Problem Statement We study the design problem associated with rods of vanishing natural curvature from a geometric and from an algorithmic perspective. To formalize this problem, we use the large-displacement small-strain model developed by Kirchhoff and Clebsch, called the *Kirchhoff rod* model for short.

The deformed state of a Kirchhoff rod can be described as a framed curve in $\mathbb{R}^3$, so a natural question to ask is *which* framed curves occur as deformed equilibrium states of rods—supposing we can freely choose the cross-sectional profile at every point. We study three flavors of this question, which differ by how much of the deformed state is prescribed, i.e., whether the twist of the rod is constrained to vanish, constrained to be a prescribed function, or not constrained at all.

Contributions Our main theoretical contribution is to show that there is a close connection between these sets of *equilibrium curves* and classical projective line geometry. This connection leads to a characterization of these curves that is both concise and computationally convenient. Furthermore, it directly translates into an algorithm that checks whether a given curve has the

equilibrium property, and lets us construct the geometry of a rod that will realize this curve as one of its equilibrium states.

This algorithm takes as input a design, consisting of a number of curves in three-dimensional space, which we would like to turn into a physical model where elastic rods take the place of these curves. For each curve, the algorithm outputs the geometry of a straight rod with a spatially-varying cross-sectional profile. This rod will adopt the shape of the input curve at static equilibrium once its endpoints are mounted in a support structure at the correct locations, orientations, and twist, as illustrated in Fig. 5.2.

The basic version of our algorithm, which applies when the gravitational effect on the deformed shape is negligible, comes with the rigorous guarantee that it terminates successfully if and only if the input is feasible. The feasibility test and geometry generation take about 10 ms, so they are fast enough to be part of an interactive design pipeline, allowing the user to receive immediate feedback upon editing a curve, for example using a spline representation.

In preparation of our main results, we formulate and show two interesting properties of the design space of Kirchhoff rods. First, we prove convexity of the set of stiffness matrices that describe the resistance to bending and twisting at a point of the rod. This indicates that many design problems involving Kirchhoff rods can be posed as convex optimization problems. Second, we investigate how using cross sections with different shapes affects the equilibrium curves that can be achieved. We prove that elliptical cross sections suffice to reach every possible deformed state, and other cross sections do not add any more design freedom.

Finally, we show how external forces and the dead load of a rod can be modeled as part of the inverse problem, thereby opening up applications in design on a larger scale. To take these forces into account, we present an iterative algorithm that post-processes the result obtained with the basic, load-free algorithm. Convergence of this algorithm is at least linear on all of our examples, as we show by empirical means.

We apply our design algorithm to several fields of application, such as fixture design, interior design, and soft robotics. In this context, we discuss our fabrication pipeline for producing physical copies of rods using 3d-printed molds and silicone casting. The efficacy of our approach is validated by comparing photographs of our manufactured examples to renderings of the target designs.

5.2 Overview

The technical sections of this paper are organized in three parts. Our contributions are presented in Sections 5.4 and 5.6-5.8, while Sections 5.3 and 5.5 serve as technical introductions.

1. Kirchhoff Rods We summarize the classical Kirchhoff rod model, with an emphasis on how to compute the stiffness matrix for a given cross section, and give a condensed derivation of the equilibrium equation for clamped-clamped boundary conditions (Section 5.3).

These tools are used to characterize the set of all stiffness matrices achievable within Kirchhoff rod theory and to show that this set is convex. Furthermore, we prove that a much smaller set of stiffness matrices, namely, that induced by elliptical cross sections, suffices to reach every possible equilibrium state (Section 5.4).

2. Equilibrium Curves We give a summary of Plücker coordinates and linear line complexes (Section 5.5), our main tools to study the design space of Kirchhoff rods. Then, we present a geometric characterization of twist-free equilibrium states, which directly generalizes the result in the plane setting, presented in Theorem 1. We explore interactive design of these curves, and discuss how to heuristically avoid stability issues (Section 5.6).

Next, we drop the twist-free constraint and present a geometric characterization of all Frénet curves that appear as deformed centerlines of Kirchhoff rods at equilibrium. This leads to a computational design algorithm that decides for a given (framed) curve whether it has the equilibrium property, and, in case it does, computes the geometry of the corresponding Kirchhoff rod (Section 5.7).

3. Design under Load We discuss a generic load model that supports point loads and line loads, such as the dead load of a rod, and the resulting modification to the equilibrium equation. The problem of finding solutions to this equation is then posed as a fixed-point problem, which can be solved via iteration (Section 5.8).

Finally, in Section 5.9, we show several physical examples that have been designed with our approach and manufactured using silicone casting, before we draw conclusions in Section 5.10.

All algorithms presented in this paper rely only on linear programs and initial value problems, both of which can be solved numerically within a few milliseconds. The result is a set of techniques fast enough to produce results within a fraction of a second, so they can be used to give immediate feedback during an interactive editing session, or as part of more complex, iterative algorithms.

5.3 Kirchhoff Rods: Technical Preliminaries

The goal of this section is to introduce the mathematical tools to describe a popular model for the deformation of thin elastic rods, developed by Kirchhoff and Clebsch. Readers familiar with this subject may skip ahead to Eq. 5.6 and consult the *cheat sheet* in Appendix B.5 for an overview of our notation.

A rod is a slender three-dimensional body whose extent in one direction is much greater than that in the orthogonal directions, such as beams, cables, yarn, and hair. It can be shown that deformations of a rod under moderate loads admit a number of kinematic simplifications, such as near-inextensibility of the centerline, and cross sections remaining nearly planar in bending. These assumptions lead to a reduction of the full three-dimensional displacement field to a one-dimensional description via curvatures and twist, which is captured mathematically by the concept of framed curves.

5.3.1 Framed Curves

We will focus our investigation on rods with a center line that is straight in the initial state. To keep track of bending and twisting modes, we rigidly attach a standard orthonormal frame (e_1, e_2, e_3) to every point of the center line, assuming that the direction of the rod coincides with e_3, as shown in Fig. 5.3 (left).

Once forces are applied to the rod, the center line deforms isometrically into an *arc-length parametrized curve* $\gamma : (0, \ell) \to \mathbb{R}^3$, with ℓ the length of the rod. Likewise, the frames

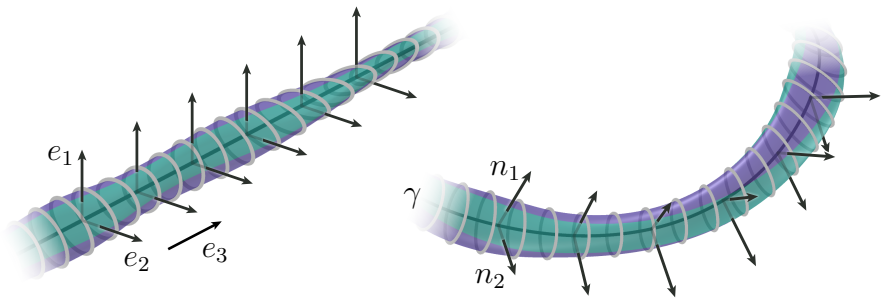


Figure 5.3: **Kirchhoff Rod.** *Left:* Initial, undeformed state. *Right:* Deformed state with bending and twist. The cross sections (gray) are anisotropic, and vary across the length. Their centroids form the centerline γ (black), to which the material normals n_1 and n_2 (black) are attached. The purple-green stripe pattern on the surface visualizes the twist in the deformed state.

rotate such that e_1 and e_2 map onto the *material normals* n_1 and n_2, and e_3 onto the *curve tangent* γ'. These three orthonormal vectors form the columns of the *moving frame* $F = (n_1, n_2, \gamma') : (0, \ell) \to SO(3)$. The pair (γ, F) is called a *framed curve*. Fig. 5.3 shows an anisotropic Kirchhoff rod in its undeformed and deformed state. In the remainder of this subsection, we recap the relevant formulas from Section 3.1.2, where they are presented in more detail.

Darboux Vector and Curvature The derivative of F wrt the *arc-length parameter* s is described geometrically by the *Darboux vector* $\omega : (0, \ell) \to \mathbb{R}^3$, which satisfies $F' = [\omega]_\times F$. Here, $[\omega]_\times$ denotes the skew-symmetric matrix such that $[\omega]_\times v = \omega \times v$ for all $v \in \mathbb{R}^3$. The coordinates of ω with respect to F are called the *curvature vector* $k : (0, \ell) \to \mathbb{R}^3$, so $\omega = Fk$. The components $k = (\kappa_1, \kappa_2, \tau)^t$ are known as the *material curvatures* κ_i, for $i = 1, 2$, and the *twist* τ.

The normal component of ω, denoted by ω_n, is the same for all frames adapted to γ, and can be computed as

$$\omega - \langle \omega, \gamma' \rangle \gamma' = \omega_n = \gamma' \times \gamma''.$$

Likewise, the *(geometric) curvature* $\kappa = \sqrt{\kappa_1^2 + \kappa_2^2} \geq 0$ is determined by γ alone, and can be computed as $\kappa = \|\gamma''\| = \|\omega_n\|$. Analogous to ω_n, we also introduce notation for the first two columns of F and the first two entries of k:

$$F_n := (n_1, n_2) = FS, \quad k_n := (\kappa_1, \kappa_2)^t = S^t k, \quad \text{with} \quad S = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}.$$

This lets us write $\omega_n = F_n k_n$ for any frame.

It will be useful to express k purely in terms of F. To do this, we can use the following identity: for any $Q \in \mathrm{SO}(3)$ and $v \in \mathbb{R}^3$, it holds that $Q[v]_\times Q^t = [Qv]_\times$.[1] Then, we compute $F'F^t = [\omega]_\times = [Fk]_\times = F[k]_\times F^t$, which implies that $[k]_\times = F^t F'$.

Special Frames A frame with $\tau \equiv 0$ is called *parallel* and describes a rod deformation without twist. For any given curve, a parallel frame F is uniquely determined by the image under F of a single point, for example $F(0)$. Two parallel frames differ only by a constant

[1] Proof: Let $w \in \mathbb{R}^3$. Then, $Q[v]_\times Q^t w = Q(v \times (Q^t w)) = (Qv) \times w = [Qv]_\times w$. $\square$

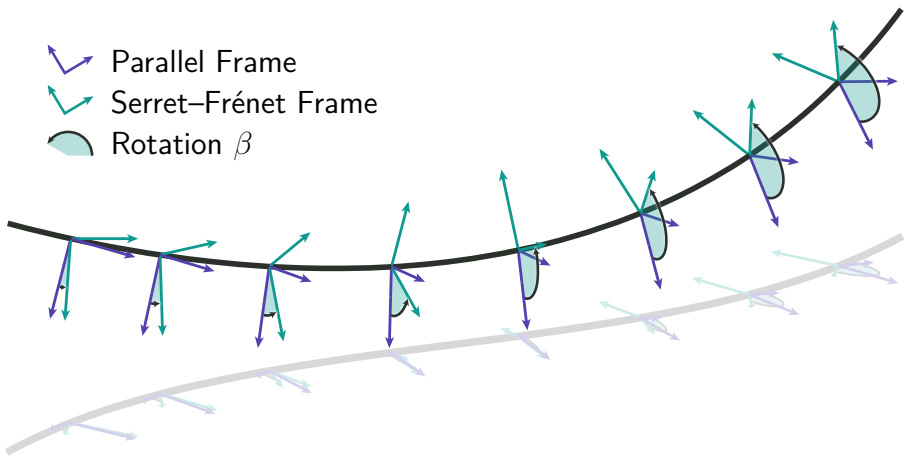


Figure 5.4: **Rotation Between a Parallel and the Serret–Frénet Frame.** Any two frames adapted to the same curve are related through a rotation with some angle $\beta : (0, \ell) \to \mathbb{R}$ around the curve tangent. In this example, β relates a parallel frame (purple) and the Frénet frame (green) of the curve.

rotation in the normal plane. The Darboux vector of a parallel frame is contained in the normal plane, so $\omega = \omega_n$.

Away from inflection points, a curve has a unique *Serret–Frénet frame* defined by the *principal normal* $n_1 = \gamma''/\|\gamma''\|$ and characterized by $\kappa_1 \equiv 0$. The curve *binormal* $n_2 = (\gamma' \times \gamma'')/\|\gamma''\|$ is parallel to ω_n. A curve that has a Serret–Frénet frame everywhere is called a *Frénet curve* and is characterized by $\kappa > 0$.

Relating Adapted Frames Two frames F and F_β adapted to the same curve γ share the same third basis vector at every point, i.e., $Fe_3 = \gamma' = F_\beta e_3$. Thus, they are related through a rotation around the third basis vector by an angle $\beta : (0, \ell) \to \mathbb{R}$ that may vary as a function of s. Likewise, we can relate their curvature vectors:

$$\begin{aligned} F_{\beta,n} = F_n Q_\beta, \quad \text{with} \quad Q_\beta = \begin{pmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{pmatrix}, \\ k_{\beta,n} = Q_\beta^t k_n, \quad \tau_\beta = \tau + \beta'. \end{aligned} \tag{5.1}$$

Fig. 5.4 shows a curve with a parallel frame and its Serret–Frénet frame, as well the rotation between the two frames.

5.3.2 Elastic Energy

The elastic energy of a deformed Kirchhoff rod is defined based on the assumption that bending and twisting modes of deformation can be decoupled and contribute separately to the energy. The contribution of each mode is derived by assuming a state of uniform bending or twisting, and computing the respective energy from three-dimensional elasticity. We will only discuss aspects of the resulting formulation that are relevant to subsequent sections—for a detailed derivation, see Audoly and Pomeau [AP10, 3.3–3.5].

The elastic energy of a Kirchhoff rod is defined as

$$W = \frac{1}{2}\int_0^\ell \langle k, Kk \rangle, \text{ with } K = \left(\begin{array}{cc|c} EI & & 0 \\ & & 0 \\ \hline 0 & 0 & \mu J \end{array}\right), \; I = \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix}. \tag{5.2}$$

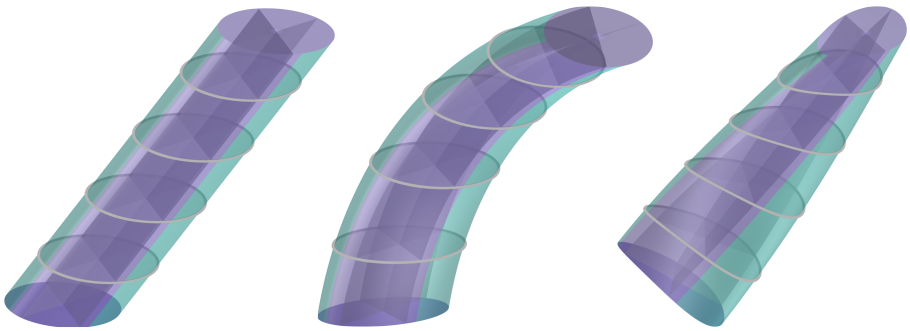


Figure 5.5: **Deformation Modes.** *Left:* Undeformed rod with cross sections and coordinate planes drawn. *Center:* Uniform bending. *Right:* Uniform twist.

The Young's modulus $E > 0$ and shear modulus $\mu > 0$ only depend on the base material of the rod, and are assumed to be constant across the length. The stiffness matrix $K(s)$ contains the bending stiffness $EI(s)$ and twisting stiffness $\mu J(s)$, with $I(s)$ and $J(s)$ dependent on the cross section of the rod at $s \in (0, \ell)$.

In particular, the bending rigidity I is the area moment of inertia tensor, whose coordinates are given by

$$I(s) = \int_{\mathcal{D}(s)} \begin{pmatrix} y^2 & -xy \\ -xy & x^2 \end{pmatrix} dA(x, y). \tag{5.3}$$

Here, $\mathcal{D}(s) \subset \mathbb{R}^2$ is the cross section of the rod at s, with the centerline passing through its centroid. The integrand can be written as the outer product of $(-y, x)^t$ with itself, so $I(s) \in S^2_{++}$, i.e., $I(s)$ is symmetric positive-definite whenever $\mathcal{D}(s)$ has positive area measure. The bending energy is given by $\frac{1}{2}E\langle k_n, Ik_n\rangle$ and is due to normal stresses away from the centerline, as seen in Fig. 5.5 (center).

If $\mathcal{D}(s)$ is simply connected, the torsional rigidity J is given by the Dirichlet energy of the solution χ to a Poisson equation:

$$J(s) = 4\int_{\mathcal{D}(s)} \|\nabla\chi\|^2, \quad \text{with} \quad \begin{array}{ll} \Delta\chi = -1 & \text{in } \mathcal{D}(s), \\ \chi = 0 & \text{on } \partial\mathcal{D}(s). \end{array} \tag{5.4}$$

The physical interpretation of χ is a potential for the out-of-plane shear strains that appear as a result of twist. In general, neither I nor J have closed-form solutions, but they do for special cases such as circular and elliptical disks. The twisting energy is given by $\frac{1}{2}\mu J\tau^2$ and is due to shear stresses away from the centerline. Twisting causes cross sections to warp out of plane to reach the minimum-energy state, as shown in Fig. 5.5 (right).

5.3.3 Equilibrium Equations

We assume *kinematic*, or double-clamped, boundary conditions: both γ and F are fixed at $s = 0$ and $s = \ell$. These boundary conditions are often encountered in architectural and interior design applications, and they make for the richest design space of equilibrium states.

To set up the variational problem, we choose F as the primary variable, so fixing F at both ends imposes Dirichlet boundary conditions. Assuming that $\gamma(0)$ coincides with the origin, we can express γ as a function of F via $\gamma(s) = \int_0^s \gamma' = \int_0^s Fe_3$. Thus, the endpoint constraint $\gamma(\ell) = \gamma_\ell$ takes the form of an integral constraint $\int_0^\ell Fe_3 = \gamma_\ell$. Constrained extremals of the

Kirchhoff energy are characterized by extremals of the Lagrangian

$$\mathcal{L} = \int_0^\ell \left(\frac{1}{2}\langle k, Kk \rangle - \langle c, Fe_3 \rangle \right), \tag{5.5}$$

with Lagrange multiplier $c \in \mathbb{R}^3$.

To derive the Euler–Lagrange equations, we first discuss admissible variations of F. Any one-parametric family of variations takes the form $\tilde{F}(s, \varepsilon)$, such that $\tilde{F}(s, 0) = F(s)$, and $\tilde{F}(s, \varepsilon) \in \mathrm{SO}(3)$ for all $s \in (0, \ell)$ and $\varepsilon \in (-\varepsilon_0, \varepsilon_0)$. To characterize the variation $\delta F(s) := (\partial/\partial\varepsilon)\tilde{F}(s, \varepsilon)|_{\varepsilon=0}$, we differentiate the equation $\tilde{F}\tilde{F}^t = \mathrm{id}$ with respect to ε. This shows that $\delta F\, F^t$ is skew-symmetric, so there exists some $\eta : (0, \ell) \to \mathbb{R}^3$ such that $\delta F = [\eta]_\times F$.

Next, we discuss variations δk induced by η. From differentiating δF with respect to s, we get $\delta F' = [\eta']_\times F + [\eta]_\times F'$, and, from skew-symmetry of $[\eta]_\times$, we arrive at $\delta F^t = -F^t[\eta]_\times$. Then, we take the variation of $[k]_\times = F^t F'$:

$$[\delta k]_\times = -F^t[\eta]_\times F' + F^t([\eta']_\times F + [\eta]_\times F') = F^t[\eta']_\times F = [F^t\eta']_\times.$$

This implies $\delta k = F^t\eta'$.

Now, we can compute the variation of the Lagrangian:

$$\delta\mathcal{L} = \int_0^\ell (\langle F^t\eta', Kk \rangle - \langle c, [\eta]_\times Fe_3 \rangle) = \int_0^\ell (\langle FKk, \eta' \rangle + \langle c \times \gamma', \eta \rangle).$$

According to the fundamental lemma of variational calculus, we have $\delta\mathcal{L} = 0$ for all test functions η if and and only if $(FKk)' = c \times \gamma'$. We can integrate this equation to arrive at the equilibrium equation

$$FKk = c \times \gamma + \bar{c}, \tag{5.6}$$

with integration constant $\bar{c} \in \mathbb{R}^3$. This shows that (γ, F) represents a static equilibrium of a Kirchhoff rod with stiffness K and kinematic boundary conditions if and only if Eq. 5.6 holds for some $c, \bar{c} \in \mathbb{R}^3$.

Our main contribution is to give geometric characterizations of (framed) curves having this property at three levels of generality, which are captured by the following definitions:

Definition 4. Let (γ, F) be an arc-length parametrized framed curve of length ℓ with curvature vector k. Assume there exist $c, \bar{c} \in \mathbb{R}^3$ and domains $\mathcal{D}(s) \subset \mathbb{R}^2$ centered at the origin for all $s \in (0, \ell)$ such that $FKk = c \times \gamma + \bar{c}$ holds, where K denotes the stiffness matrix induced by $\mathcal{D}$. Then, (γ, F) is called a *framed equilibrium curve*.

Definition 5. Let γ be an arc-length parametrized curve. If there exists a frame $F : (0, \ell) \to \mathrm{SO}(3)$ adapted to γ such that (γ, F) is a framed equilibrium curve, then γ is called an *equilibrium curve*. If, additionally, F can be chosen to be a parallel frame, then γ is called a *parallel equilibrium curve*.

Our geometric characterizations of parallel equilibrium curves, equilibrium curves, and framed equilibrium curves are given in Sections 5.6.1, 5.7.1, and 5.7.5, respectively.

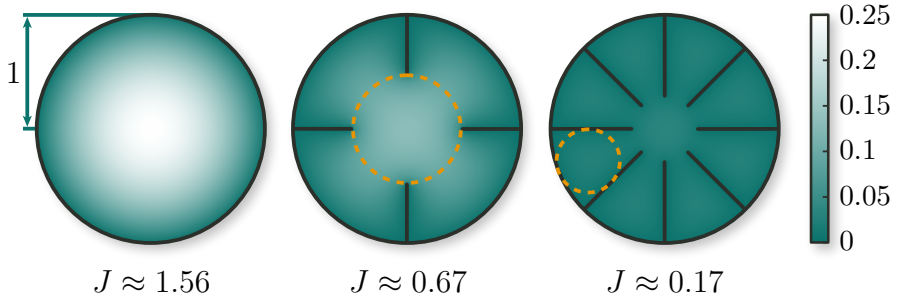


Figure 5.6: J **has no positive lower bound.** The green gradient shows the solution χ to Eq. 5.4 for circular disks with an increasing number and length of cuts. Progressively adding cuts makes J arbitrarily small while maintaining I. Greatest incircles, as used in Appendix B.1, are drawn in yellow.

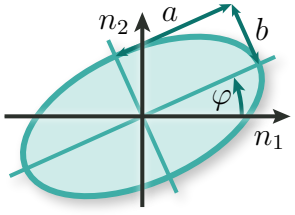
5.4 Kirchhoff Rods: The Constitutive Relation

To uniquely define the geometry of a straight elastic rod, we have to choose a cross section $\mathcal{D}(s) \subset \mathbb{R}^2$ at every $s \in (0, \ell)$, such that the center line passes through the centroid of $\mathcal{D}(s)$. This choice determines $K(s)$ and thus the mechanical behavior of the rod at this point. Ultimately, we want to study the set of equilibrium states, but to do this, we first have to characterize the set of all stiffness matrices K that are induced by admissible cross sections.

We carry out this characterization in Section 5.4.1, and then infer two results that are relevant in the context of design. First, we show in Section 5.4.2 that the set of admissible stiffness matrices is convex, thus allowing design problems with convex objectives to be cast as convex optimization problems. Second, we prove in Section 5.4.3 that the subset of admissible stiffness matrices corresponding to elliptical cross sections is sufficient to span the entire design space of framed equilibrium curves. Thus, we do not lose any design freedom upon neglecting exotic cross sections that would be hard to fabricate or fail to satisfy the Kirchhoff assumptions.

5.4.1 Admissible Stiffness Matrices

We have already seen in Eqs. 5.2–5.4 that every stiffness matrix K consists of a 2-by-2 block $EI \in S^2_{++}$, the bending stiffness, and $\mu J > 0$, the torsional stiffness. But is every matrix that obeys these constraints induced by an admissible cross section?



Regarding the bending stiffness, it is easy to see that every $I \in S^2_{++}$ can be attained, for example using elliptical cross sections. In particular, we can choose the radii a, b of an ellipse to determine the eigenvalues of I and the orientation φ to determine the eigenvectors, as shown in the inset. The resulting area moment matrix is

$$I = \frac{\pi}{4} Q \begin{pmatrix} ab^3 & 0 \\ 0 & a^3 b \end{pmatrix} Q^t, \quad \text{with} \quad Q = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix}. \tag{5.7}$$

For the torsional stiffness, we see from Eq. 5.4 that any $J > 0$ can be attained in principle, for example by circular disks of different radii. However, if we restrict our attention to cross sections with fixed I, the range of attainable J is limited by

$$J \leq 4\psi(I), \quad \text{with} \quad \psi : S^2_{++} \to \mathbb{R} : X \mapsto \frac{\det X}{\operatorname{tr} X}, \tag{5.8}$$

as shown by Diaz and Weinstein [DW48, p. 5]. This upper bound on J is tight, and the unique maximizer for any given I is an ellipse. We can see from Eq. 5.7 that in this case, $J = \frac{\pi a^3 b^3}{a^2+b^2}$.

On the other hand, we show that there is no positive lower bound:

Proposition 2. *Given $I \in S^2_{++}$ and $J > 0$, there is a bounded domain $\mathcal{D} \subset \mathbb{R}^2$ with bending rigidity I and torsional rigidity J if and only if (I, J) is an element of*

$$\mathcal{K} = \{(I, J) \in S^2_{++} \times \mathbb{R} : 0 < J \leq 4\psi(I)\}. \tag{5.9}$$

Proof. See Appendix B.1 for a rigorous proof of the tightness of the lower bound on J for any fixed I. □

A heuristic argument that we can decrease J without changing I is illustrated in Fig. 5.6: We start with a cross section realizing I, and progressively add cuts from the boundary to the interior of the domain. This does not change I, because we only remove zero-measure sets from $\mathcal{D}$. However, J can be made arbitrarily small as the boundary points of the domain move closer and closer together.

We assume E and μ to be fixed, so a pair $(I, J) \in \mathcal{K}$ uniquely defines a stiffness matrix $K = \operatorname{diag}(EI, \mu J)$, and vice versa. For convenience, we will sometimes write $K = (I, J)$ and $K \in \mathcal{K}$ by abuse of notation.

5.4.2 Convexity of $\mathcal{K}$

Designing a Kirchhoff rod with a prescribed equilibrium state (γ, F) amounts to finding $c, \bar{c} \in \mathbb{R}^3$ and $K : (0, \ell) \to \mathcal{K}$ that solve Eq. 5.6. Because Eq. 5.6 is linear in $c, \bar{c}$, and K, we need only convexity of $\mathcal{K}$ to show that this is a convex problem for arbitrary convex objectives. Our proof is based on:

Lemma 3. *The function $\psi : X \mapsto \frac{\det X}{\operatorname{tr} X}$ is concave on S^2_{++}.*

Proof. Let $X, Y \in S^2_{++}$ and $t \in (0, 1)$. We need to show that $(1-t)\psi(X) + t\psi(Y) \leq \psi((1-t)X + tY)$, which expands to

$$\frac{(1-t)\operatorname{tr} Y \det X + t \operatorname{tr} X \det Y}{\operatorname{tr} X \operatorname{tr} Y} \leq \frac{\det((1-t)X + tY)}{(1-t)\operatorname{tr} X + t \operatorname{tr} Y}.$$

We multiply through by the product of the denominators, which is strictly positive, expand $\det((1-t)X + tY)$ in terms of the components of X and Y, and divide by $t(1-t)$, which is also positive. Most terms cancel, and we arrive at the equivalent statement

$$(\operatorname{tr} X)^2 \det Y + (\operatorname{tr} Y)^2 \det X \leq \operatorname{tr} X \operatorname{tr} Y (X_{11}Y_{22} - 2X_{12}Y_{12} + X_{22}Y_{11}),$$

which can be factorized to give

$$0 \leq (X_{11}Y_{22} - X_{22}Y_{11})^2 + (Y_{12} \operatorname{tr} X - X_{12} \operatorname{tr} Y)^2.$$

This shows that ψ is concave. □

Proposition 4. *The set $\mathcal{K}$ is convex.*

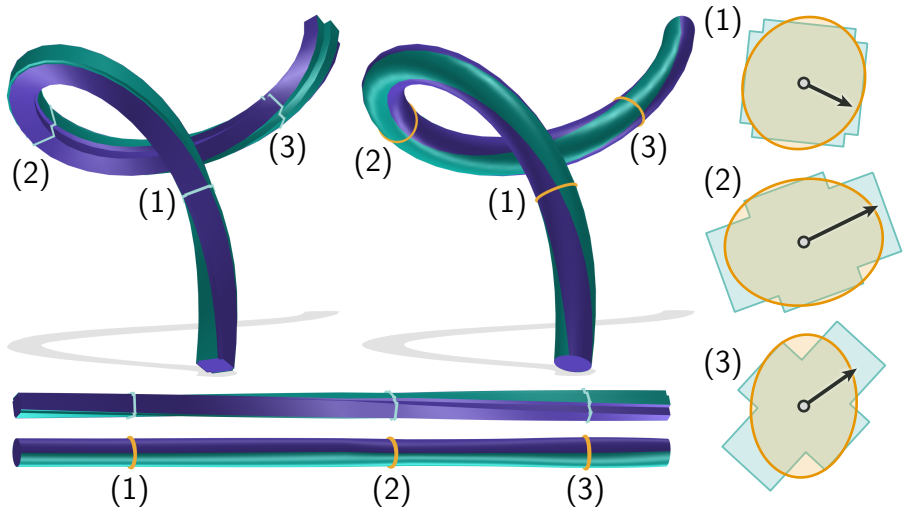


Figure 5.7: **Elliptification.** *Left:* Rod with cross-shaped cross sections (light blue), and rod with elliptical cross sections (yellow) having the same equilibrium state. *Right:* Comparison between cross-shaped and elliptical cross sections at marked locations, with k_n visualized as arrow.

Proof. Let $(I_0, J_0), (I_1, J_1) \in \mathcal{K}$, and $t \in (0, 1)$. Because $I_0, I_1 \in S^2_{++}$, and S^2_{++} is convex, we have $(1-t)I_0 + tI_1 \in S^2_{++}$. From concavity of ψ, it follows that

$$0 < (1-t)J_0 + tJ_1 \leq 4((1-t)\psi(I_0) + t\psi(I_1)) \leq 4\psi((1-t)I_0 + tI_1),$$

which shows that $((1-t)I_0 + tI_1, (1-t)J_0 + tJ_1) \in \mathcal{K}$. □

This result guarantees that we can numerically find the global optimizer of optimization problems over $\mathcal{K}$ constrained by Eq. 5.6, as long as the objective function is convex. In particular, we can solve the constraint satisfaction problem (CSP) of determining whether a given framed curve is a framed equilibrium curve. In Section 5.7, we improve this result and show that the CSP can even be solved by a linear program.

5.4.3 Elliptical Cross Sections

While $\mathcal{K}$ is convenient for numerical optimization, it contains stiffness matrices that are only induced by cross sections like the one in Fig. 5.6 (right), which are impractical to manufacture. Moreover, cross sections like this will buckle even under moderate loads, and thus break the assumptions of Kirchhoff rods. Therefore, it would be best to avoid using cross sections with $J \ll 4\psi(I)$ in design.

This raises two questions: How is the design space of Kirchhoff rods reduced if we restrict ourselves to a proper subset of $\mathcal{K}$ that has impractical cross sections removed? And what is a good subset to use? Surprisingly, a very restrictive choice is still optimal:

$$\mathcal{K}^* := \partial\mathcal{K} \cap \mathcal{K} = \{(I, J) \in S^2_{++} \times \mathbb{R} : J = 4\psi(I)\}, \tag{5.10}$$

which contains exactly the stiffness matrices induced by elliptical cross sections, is enough to realize all framed equilibrium curves, with no reduction of the design space, as we show here:

Proposition 5. *Let (γ, F) be a framed equilibrium curve such that $FKk = c \times \gamma + \bar{c}$ holds with $c, \bar{c} \in \mathbb{R}^3$ and $K : (0, \ell) \to \mathcal{K}$. Then, there exists $K^* : (0, \ell) \to \mathcal{K}^*$ such that $FK^*k = c \times \gamma + \bar{c}$.*

Proof. We rewrite the equilibrium equation by splitting it up into its normal and tangential components, which gives

$$EIk_n = F_n^t(c \times \gamma + \bar{c}), \quad \mu J\tau = \langle \gamma', c \times \gamma + \bar{c} \rangle. \tag{5.11}$$

To show the statement, we compute $I^* \in S^2_{++}$ for each $s \in (0, \ell)$ such that $(I^*, J) \in \mathcal{K}^*$ and $I^* k_n = I k_n$.

Let $Q \in SO(2)$ such that $Qk_n = \kappa e_1$, and define $\tilde{I} = QIQ^t$. Let

$$\tilde{I}^* = \begin{pmatrix} \tilde{I}_{xx} & \tilde{I}_{xy} \\ \tilde{I}_{xy} & \tilde{I}^*_{yy} \end{pmatrix}, \quad \text{with} \quad \tilde{I}^*_{yy} = \frac{J\tilde{I}_{xx} + 4\tilde{I}^2_{xy}}{4\tilde{I}_{xx} - J},$$

which only has its bottom-right entry different from $\tilde{I}$. Then, $I^* = Q^t \tilde{I}^* Q$ satisfies all requirements, as we will now show.

The equilibrium equation is satisfied because

$$I^* k_n = Q^t \tilde{I}^* Q k_n = \kappa Q^t \tilde{I}^* e_1 = \kappa Q^t \tilde{I} e_1 = Q^t \tilde{I} Q k_n = I k_n.$$

To verify that $I^* \in S^2_{++}$, it suffices to show that $\tilde{I}^* \in S^2_{++}$. We have $\tilde{I}_{xx} > 0$ because I and thus $\tilde{I}$ is in S^2_{++}. Furthermore, $\det \tilde{I}^* = \frac{J(\tilde{I}^2_{xx} + \tilde{I}^2_{xy})}{4\tilde{I}_{xx} - J} > 0$ if $J < 4\tilde{I}_{xx}$. But this always holds because

$$J \leq 4\psi(I) = 4\psi(\tilde{I}) = \frac{4(\tilde{I}_{xx}\tilde{I}_{yy} - \tilde{I}^2_{xy})}{\tilde{I}_{xx} + \tilde{I}_{yy}} \leq 4\tilde{I}_{xx} \frac{\tilde{I}_{yy}}{\tilde{I}_{xx} + \tilde{I}_{yy}} < 4\tilde{I}_{xx},$$

where we have used that ψ only depends on the tensor invariants of its argument. Finally, $J = 4\psi(I^*) = 4\psi(\tilde{I}^*)$, and thus $(I^*, J) \in \mathcal{K}^*$ is shown by direct computation, from the definition of $\tilde{I}^*$. □

This result shows that any Kirchhoff rod equilibrium state is also attainable by a Kirchhoff rod consisting exclusively of elliptical cross sections. Fig. 5.7 shows an example in which a rod with spatially-varying cross-shaped cross sections is converted to a rod with elliptical cross sections. Even though the resulting elliptical cross sections do not have the same bending rigidity, it is guaranteed that the equilibrium state is maintained.

Using $\mathcal{K}$ in optimization and in proofs, and $\mathcal{K}^*$ for design and fabrication gives us the best of both worlds: The convexity of $\mathcal{K}$ gives optimality guarantees in optimization and simplifies mathematical analysis. Meanwhile, any solution obtained using $\mathcal{K}$ can be converted to a solution in $\mathcal{K}^*$, which avoids cross-sectional buckling issues, and is guaranteed to yield moldable geometries because elliptical cross sections are convex.

5.5 Equilibrium Curves: Technical Preliminaries

Our next goal is to give a thorough analysis of the types of equilibrium curves introduced in Definitions 4 and 5. In particular, we provide characterizations that relate the shape of these curves to existing concepts from projective line geometry, and that pave the way to very efficient computational design algorithms. We will show that some fundamental inverse design problems related to Kirchhoff rods can be solved exactly, in the sense that we can decide computationally whether there is a solution for a given input, and find a solution near-instantaneously if it exists.

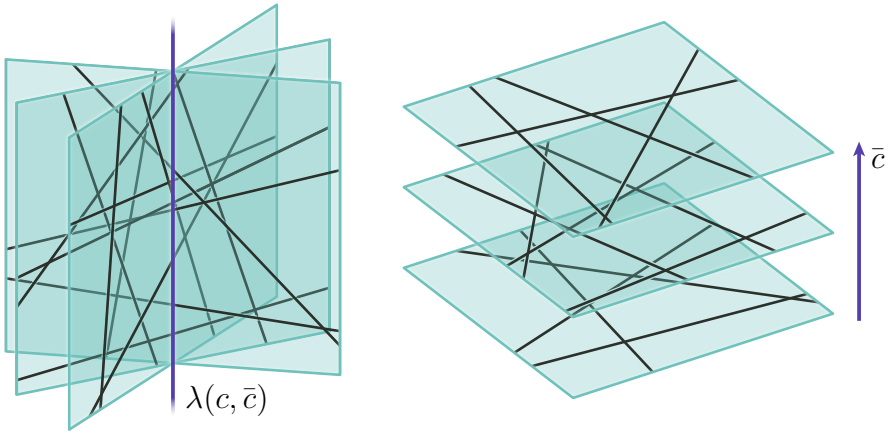
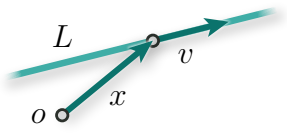


Figure 5.8: **Singular Linear Complexes.** *Left:* A Euclidean singular linear complex ($c \neq 0$) contains all lines (black) incident to a fixed Euclidean line (purple) with Plücker coordinates $(c, \bar{c})$. *Right:* An ideal singular linear complex ($c = 0$) contains all lines (black) orthogonal to a fixed direction $\bar{c}$ (purple).

Looking back at the previous section, we found that choosing domains $\mathcal{D}(s) \subset \mathbb{R}^2$ in Definition 4 is equivalent to choosing a stiffness function $K : (0, \ell) \to \mathcal{K}$. Proposition 5 further guarantees that we can restrict $\mathcal{K}$ to $\mathcal{K}^*$, thus using only elliptical cross sections, without changing the design space of Kirchhoff rod equilibrium states. To characterize this space, we ask: For which (framed) curves is it possible to choose $c, \bar{c} \in \mathbb{R}^3$ and a stiffness function so Eq. 5.6 is satisfied? Our answers, which we give in Sections 5.6 and 5.7, rely on concepts from the geometry of lines in $\mathbb{R}^3$, and in particular on linear line complexes, which we introduce below. For a more extensive treatment, we point the reader to Pottmann and Wallner [PW01, Ch. 2.1, 3.1].

5.5.1 Line Geometry



Let L be a line in $\mathbb{R}^3$, with direction $v \in \mathbb{R}^3$ and passing through a point $x \in \mathbb{R}^3$. We call $(l, \bar{l}) := (v, x \times v) \in \mathbb{R}^6$ the *Plücker coordinates* of L, and write $L = \lambda(l, \bar{l})$ to denote the line defined by $(l, \bar{l})$ as a subset of $\mathbb{R}^3$. Plücker coordinates satisfy $\langle l, \bar{l} \rangle = 0$ and are homogeneous, i.e., every non-zero scalar multiple refers to the same line. The projective extension of this set is known as the Klein quadric

$$\Lambda_{\text{kl}} := \{(l, \bar{l}) \in \mathbb{P}^5 : \langle l, \bar{l} \rangle = 0\}.$$

Plücker coordinates of the form $(0, \bar{l})$ represent an ideal line, which contains exactly the ideal points ("points at infinity") with directions orthogonal to $\bar{l}$. One useful application of Plücker coordinates is the computation of intersections: two lines $\lambda(l, \bar{l})$ and $\lambda(c, \bar{c})$ intersect if and only if $\langle l, \bar{c} \rangle + \langle \bar{l}, c \rangle = 0$.

The set of all lines whose Plücker coordinates satisfy a homogeneous linear equation with (fixed) coefficients $c, \bar{c} \in \mathbb{R}^3$,

$$\mathcal{C} := \{(l, \bar{l}) \in \Lambda_{\text{kl}} : \langle l, \bar{c} \rangle + \langle \bar{l}, c \rangle = 0\},$$

is known as a *linear line complex*. If $(c, \bar{c}) \in \Lambda_{\text{kl}}$, i.e., if $(c, \bar{c})$ are themselves Plücker coordinates of a line, the complex is said to be *singular* and consists of all lines intersecting $\lambda(c, \bar{c})$, as

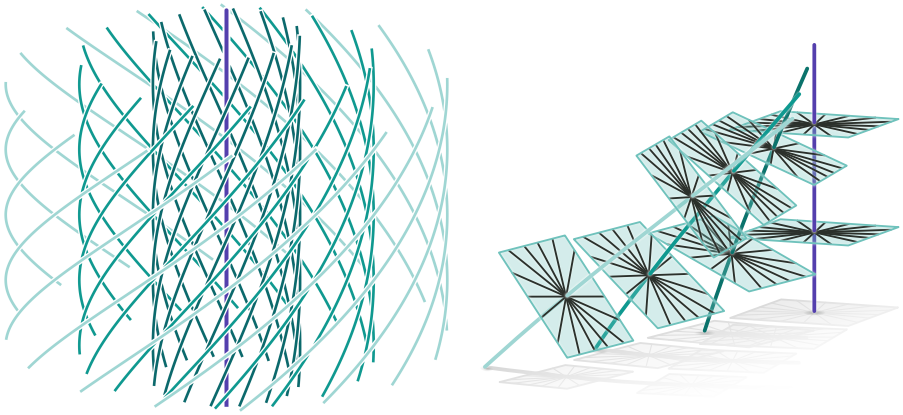


Figure 5.9: **Regular Linear Complexes.** *Left:* A helical motion is a family of helices (green) with fixed axis (purple) and fixed pitch. *Right:* For every helix (green) in the helical motion and its axis (purple), take the pencil of normal lines (black) at every point. The union of all such line pencils forms a regular linear complex.

shown in Fig. 5.8 (left). In case $(c, \bar{c})$ is ideal, so $c = 0$, the complex $\mathcal{C}$ consists of all lines with directions orthogonal to $\bar{c}$, see Fig. 5.8 (right).

We can also interpret a linear complex geometrically if $(c, \bar{c}) \notin \Lambda_{\text{kl}}$, in which case $\mathcal{C}$ is called *regular*. To do this, consider the vector field $h : x \mapsto c \times x + \bar{c}$ in $\mathbb{R}^3$. A vector field of this form is called *helical*, because every field line, i.e., every curve tangent to h in every point, is a helix. The family of all field lines of h is called a *helical motion* and consists of all helices with a fixed axis and a fixed pitch, see Fig. 5.9 (left). Next, we consider at every $x \in \mathbb{R}^3$ the pencil of lines through x that are normal to the helix passing through this point, see Fig. 5.9 (right). The union of all such line pencils gives exactly the lines contained in $\mathcal{C}$. In summary: A regular linear complex is the set of *path normals of a helical motion*.

5.6 Parallel Equilibrium Curves

Our first application of linear line complexes is a geometric characterization of parallel equilibrium curves—curves that appear as twist-free equilibria of Kirchhoff rods. This is a natural starting point, because the result directly generalizes the characterization of plane elastic curves, which we summarize here for convenience:

Theorem 6. *Let $\gamma : (0, \ell) \to \mathbb{R}^2$ with signed curvature κ and a finite number of inflection points, and let $a \in \mathbb{R}^2$, $b \in \mathbb{R}$ not all zero. Then, there exists $K : (0, \ell) \to \mathbb{R}$ with $0 < \inf K \leq \sup K < \infty$ such that $K\kappa = \langle a, \gamma \rangle + b$ if and only if*

1. *the line $L = \{x \in \mathbb{R}^2 : \langle a, x \rangle + b = 0\}$ intersects γ exactly in its inflection points, with all intersections non-tangential;*
2. *$\kappa'(s_0) \neq 0$ at all inflection points $s_0 \in (0, \ell)$.*

In Proposition 7 below, this theorem corresponds exactly to the special case in which the constants from the right-hand side of the equilibrium equation $FKk = c \times \gamma + \bar{c}$ define a singular complex $(c, \bar{c})$. Then, the line $\lambda(c, \bar{c})$ plays the role of L in Theorem 6. In contrast, regular complexes $(c, \bar{c})$ correspond exactly to parallel equilibrium curves that are not plane.

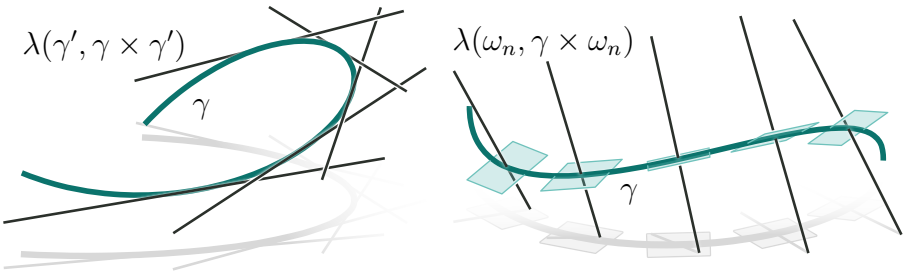


Figure 5.10: **Tangent and Binormal Lines.** *Left:* Curve (green) and its family of tangent lines (black). *Right:* Curve (green) and its family of osculating planes (green) and binormal lines (black).

Before stating the characterization theorem, we outline the remainder of this section: The class of parallel equilibrium curves proves to be rigid in the sense that only one curvature can be chosen independently at every point, unlike general curves in $\mathbb{R}^3$, which have two independent curvatures. In consequence, parallel equilibrium curves cannot be represented as splines, which makes conventional curve drawing tools ill-suited for their design. As an alternative, we derive a second characterization in Section 5.6.2, as solutions to a family of ordinary differential equations, which allows user control by directly modifying the geometric curvature of the curve. In this context, we also show that all parallel equilibrium curves are essentially warped double helices, which gives an intuitive understanding of the shapes achievable within this class. Finally, we discuss how to avoid unstable solutions by judiciously choosing cross sections.

5.6.1 Geometric Characterization

The central objects in the characterization are the *tangent lines* and *binormal lines* of a curve, i.e., the lines passing through $\gamma(s)$ in directions $\gamma'(s)$ and $\omega_n(s) = \gamma'(s) \times \gamma''(s)$ respectively, as shown in Fig. 5.10. Binormal lines are not defined at inflection points, and this is reflected in the fact that non-plane parallel equilibrium curves are always Frénet curves, as we will show. In this sense, planar solutions are exceptional: They are the only parallel equilibrium curves that may contain inflections.

Proposition 7. *Let γ be an arc-length parametrized curve in $\mathbb{R}^3$ with a finite number of inflection points. Then, γ is a parallel equilibrium curve if and only if one of the following holds:*

1. *The curve γ is not plane, and there is a linear complex $\mathcal{C}$ that contains all tangent lines of γ, but none of its binormal lines.*
2. *The curve γ is plane and satisfies the assumptions of Theorem 6.*

Solutions of type (1) *and* (2) *correspond to regular and singular values of $(c, \bar{c})$ in Eq. 5.6, respectively. Solutions of type* (1) *are Frénet curves.*

Proof. We can rewrite Eq. 5.6 as $EF_nIk_n + \mu J\tau\gamma' = c \times \gamma + \bar{c}$.

"⇒": By assumption, there exist a parallel frame F adapted to γ, constants $c, \bar{c} \in \mathbb{R}^3$, and $I : (0, \ell) \to S^2_{++}$ such that $EF_nIk_n = c \times \gamma + \bar{c}$, because $\tau \equiv 0$. Taking the inner product with γ' implies

$$0 = \langle \gamma', c \times \gamma + \bar{c} \rangle = \langle \gamma', \bar{c} \rangle + \langle \gamma \times \gamma', c \rangle,$$

so the linear complex $\mathcal{C}$ associated with $(c, \bar{c})$ contains all tangent lines of γ, with Plücker coordinates $(\gamma', \gamma \times \gamma')$. The normal part of the equilibrium equation gives $EIk_n = F_n^t(c \times \gamma + \bar{c})$, so

$$\begin{aligned} 0 \leq \langle k_n, F_n^t(c \times \gamma + \bar{c}) \rangle &= \langle F_n k_n, F_n F_n^t(c \times \gamma + \bar{c}) \rangle \\ &= \langle \omega_n, c \times \gamma + \bar{c} \rangle = \langle \omega_n, \bar{c} \rangle + \langle \gamma \times \omega_n, c \rangle, \end{aligned} \tag{5.12}$$

where the inequality follows from $I \in S^2_{++}$. Because I has full rank, we have $k_n = 0$ exactly where $F_n^t(c \times \gamma + \bar{c}) = 0$, which is equivalent to $c \times \gamma + \bar{c} = 0$ because $c \times \gamma + \bar{c}$ is in the column space of F_n.

(1) Assume $\mathcal{C}$ is regular, so $\langle c, \bar{c} \rangle \neq 0$. The column space of $[c]_\times$ is exactly the orthogonal complement of c, so there are no curve points satisfying $c \times \gamma + \bar{c} = 0$, and in consequence no points with $k_n = 0$. Thus, γ is a Frénet curve. This also shows $0 < \langle \omega_n, \bar{c} \rangle + \langle \gamma \times \omega_n, c \rangle$, so no binormal line is contained in $\mathcal{C}$.

For the sake of contradiction, assume that γ was plane. Then all tangent lines of γ are coplanar, and we can choose three that are not concurrent in a point. However, no such set of three lines is contained in a regular linear complex, so γ must not be plane.

(2) Assume $\mathcal{C}$ is singular, so $\langle c, \bar{c} \rangle = 0$. If $c = 0$, then all tangent lines of γ are orthogonal to $\bar{c}$, and γ must be plane. From $c \times \gamma + \bar{c} = \bar{c} \neq 0$, we see that $k_n \neq 0$, so γ has no inflection points. The case $c \neq 0$ reduces exactly to Theorem 6, and we show this in Appendix B.2.

"$\Leftarrow$": Assume either (1) or (2) holds, and let F be a parallel frame adapted to γ. For (2), there is nothing to prove because the result follows directly from Theorem 6. If (1) holds for some linear complex defined by $c, \bar{c} \in \mathbb{R}^3$, then we have $0 = \langle \gamma', c \times \gamma + \bar{c} \rangle$. We also have $0 < \langle \omega_n, c \times \gamma + \bar{c} \rangle$, after possibly replacing $(c, \bar{c})$ with $(-c, -\bar{c})$, because $\omega_n(s)$ and $c \times \gamma(s) + \bar{c}$ are continuous. This implies $0 < \langle k_n, F_n^t(c \times \gamma + \bar{c}) \rangle$, so we can find $I : (0, \ell) \to S^2_{++}$ such that $EIk_n = F_n^t(c \times \gamma + \bar{c})$. This implies $EF_n I k_n = c \times \gamma + \bar{c}$. □

As we have seen in the proof, the geometric conditions on tangent and binormal lines are equivalent to

$$\langle \gamma', c \times \gamma + \bar{c} \rangle = 0, \tag{5.13a}$$

$$\langle \gamma' \times \gamma'', c \times \gamma + \bar{c} \rangle > 0, \tag{5.13b}$$

for some $c, \bar{c} \in \mathbb{R}^3$. The parallel frame F does not appear in these conditions, so we need not explicitly compute it to check if they are satisfied. Furthermore, the conditions are linear in the unknowns c and $\bar{c}$, so they can be checked with a linear program in principle.

However, Eq. 5.13a is a pointwise equality constraint on γ, which means that the number of independent curvatures of γ is reduced from two to one. Thus, a manually designed curve has almost no chance of being a parallel equilibrium curve unless Eq. 5.13a is computationally enforced during the design process. In particular, non-plane curves satisfying this equation have no spline-based representation, which rules out most curve-drawing tools. Below, we propose a different way of exploring the design space of parallel equilibrium curves, but we will find another use for Eq. 5.13a in Section 5.7.2 for the design of equilibrium curves that need not be parallel.

5.6.2 ODE Characterization

We can derive two more characterizations of non-plane parallel equilibrium curves from Eq. 5.13:

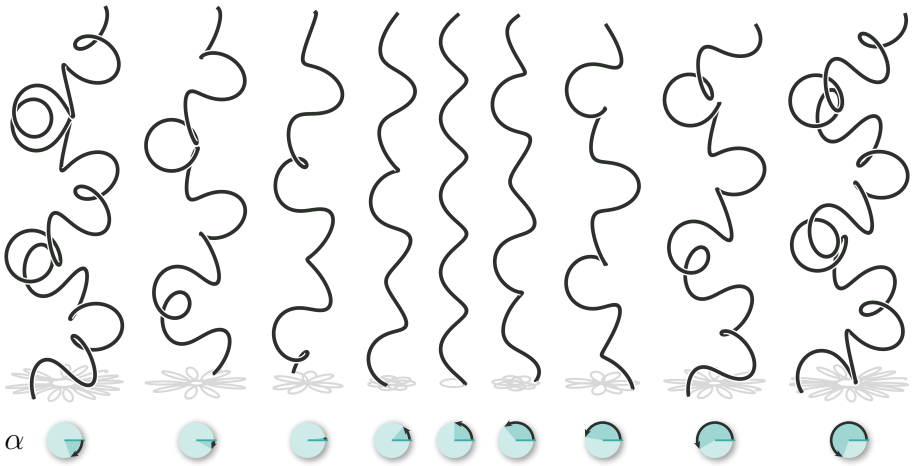


Figure 5.11: **Constant-Curvature Parallel Equilibrium Curves.** The one-parametric family of solutions to Eqs. 5.14 and 5.15 for $r = \frac{1}{2} = p$ and $\kappa \equiv 1$, obtained by sweeping $\alpha \in (0, 2\pi]$. The family interpolates smoothly between double helices and a special single-helix solution at $\alpha = \pi/2$.

Proposition 8. *Let γ be an arc-length parametrized curve in $\mathbb{R}^3$. Then the following are equivalent:*

1. *There exist $c, \bar{c} \in \mathbb{R}^3$ such that γ satisfies Eq. 5.13.*
2. *There exist $c, \bar{c} \in \mathbb{R}^3$ such that the ordered set $\{\gamma', \gamma'', c \times \gamma + \bar{c}\}$ is a right-handed orthogonal basis at every point.*
3. *There exist $c, \bar{c} \in \mathbb{R}^3$ and $m : (0, \ell) \to \mathbb{R}_{>0}$ such that*

$$\gamma''(s) = m(s) \cdot (c \times \gamma(s) + \bar{c}) \times \gamma'(s) \tag{5.14}$$

and $\langle \gamma'(0), c \times \gamma(0) + \bar{c} \rangle = 0$. It holds that $\kappa = m \cdot \|c \times \gamma + \bar{c}\|$.

Proof. (1) ⇒ (2): By differentiating $\langle \gamma', \gamma' \rangle = 1$, we get $\langle \gamma', \gamma'' \rangle = 0$, and by differentiating Eq. 5.13a,

$$0 = \langle \gamma'', c \times \gamma + \bar{c} \rangle + \langle \gamma', c \times \gamma' \rangle = \langle \gamma'', c \times \gamma + \bar{c} \rangle.$$

Eq. 5.13a and these two new equations give the three orthogonality conditions. Eq. 5.13b shows right-handedness.

(2) ⇒ (3): Eq. 5.14 with $m > 0$ and the initial condition are immediately implied by the right-handed orthogonal basis assumption. Using $\langle \gamma', c \times \gamma + \bar{c} \rangle = 0$, we compute

$$\kappa = \|\gamma' \times \gamma''\| = m \cdot \|\gamma' \times ((c \times \gamma + \bar{c}) \times \gamma')\| = m \cdot \|c \times \gamma + \bar{c}\|.$$

(3) ⇒ (1): To show that $\langle \gamma', c \times \gamma + \bar{c} \rangle$ remains constant, compute

$$\langle \gamma', c \times \gamma + \bar{c} \rangle' = \langle \gamma'', c \times \gamma + \bar{c} \rangle = m \cdot \langle (c \times \gamma + \bar{c}) \times \gamma', c \times \gamma + \bar{c} \rangle = 0,$$

so Eq. 5.13a follows from the initial condition. Eqs. 5.13a and 5.14 imply Eq. 5.13b because γ' and $c \times \gamma + \bar{c}$ are both non-zero. □

Eq. 5.14 is well-suited for the interactive exploration of parallel equilibrium curves because the user can directly modify κ, which results in a predictable change of the curve shape. A useful way of thinking about the space of feasible designs is that parallel equilibrium curves are *warped double helices*: Constant-curvature solutions of Eq. 5.14 constitute a family of double helices (with an exact discrete helical symmetry, as discussed in Appendix B.3) and contain regular helices as a special case, as shown in Fig. 5.11. Circles and lines are obtained in the limit as $\langle c, \bar{c}\rangle \to 0$ and $\kappa \to 0$. By modifying κ to be non-constant, the user can warp portions of the double helix to explore the shape space.

Apart from κ, we can also choose c, $\bar{c}$, and initial conditions $\gamma(0)$ and $\gamma'(0)$ satisfying $\langle \gamma'(0), c \times \gamma(0) + \bar{c}\rangle = 0$. After eliminating rigid motions, these reduce to three scalars $p, r, \alpha \in \mathbb{R}$ such that

$$c = e_3, \quad \bar{c} = pe_3, \quad \gamma(0) = re_1, \quad \gamma'(0) = Q_\alpha e_1, \tag{5.15}$$

where $Q_\alpha \in SO(3)$ is the rotation by angle α around the axis in direction $re_2 + pe_3$. Modifying these scalars controls the ratio between the outer and inner radius of the double helix, as well as the tightness of the windings.

5.6.3 Cross Sections & Stability

The last step in turning a parallel equilibrium curve into fabricable geometry is to choose cross sections satisfying the equilibrium equation. Because $\tau \equiv 0$, the torsional rigidity J drops out of the equation, so we need only choose $I \in S^2_{++}$ at every point to satisfy $EIk_n = F_n^t(c \times \gamma + \bar{c})$. We can then convert I into a cross section $\mathcal{D} \subset \mathbb{R}^2$, for example an elliptical one. Naturally, we can scale all cross sections uniformly along the rod to control the overall thickness. This is because a rescaled cross section $t\mathcal{D}$ has bending rigidity $t^4 I$, which solves the equilibrium equation with $(t^4 c, t^4 \bar{c})$.

To characterize all $I \in S^2_{++}$ that solve $EIk_n = F_n^t(c \times \gamma + \bar{c})$, note that k_n and $F_n^t(c \times \gamma + \bar{c})$ are parallel: From Proposition 8(2), we have that $\omega_n = \gamma' \times \gamma''$ is parallel to $c \times \gamma + \bar{c}$, and we know that $k_n = F_n^t \omega_n$. The factor of proportionality is given by m, as seen from $\|k_n\| = \kappa = m \cdot \|c \times \gamma + \bar{c}\|$. Thus, we can parametrize all admissible I by the spectral decomposition

$$I = \lambda_1 \, v_1 \otimes v_1 + \lambda_2 \, v_2 \otimes v_2, \quad \text{with}$$
$$\lambda_1 = 1/(Eq), \quad v_1 = k_n/\kappa, \quad v_2 = (-\kappa_2, \kappa_1)^t/\kappa,$$

and free parameter $\lambda_2 > 0$. If we consider an elliptical cross section in the coordinate system spanned by n_1 and n_2, the eigenvectors v_1 and v_2 give the semiaxes of the ellipse. The bending axis has direction k_n and coincides with v_1.

Even though all choices for $\lambda_2 > 0$ will give an equilibrium state, they differ greatly in their stability properties. This is because a rod will bend easily around its weak axis, but trying to bend it around its strong axis will usually result in loss of stability by buckling. It is thus essential to pick cross sections in such a way that v_1 coincides with the *major* semiaxis of the ellipse, as shown in Fig. 5.12. This is achieved by choosing $a > b$ with the notation of Eq. 5.7, which is equivalent to $\lambda_2 > \lambda_1$. Violating this rule will almost surely result in an unstable equilibrium state.

We expose the ratio $u = a/b$ to the user, and restrict its domain to $u \geq 1$. In computation, the ratio is achieved by setting $\lambda_2 = u^2 \lambda_1$. Even though this rule excludes a common source

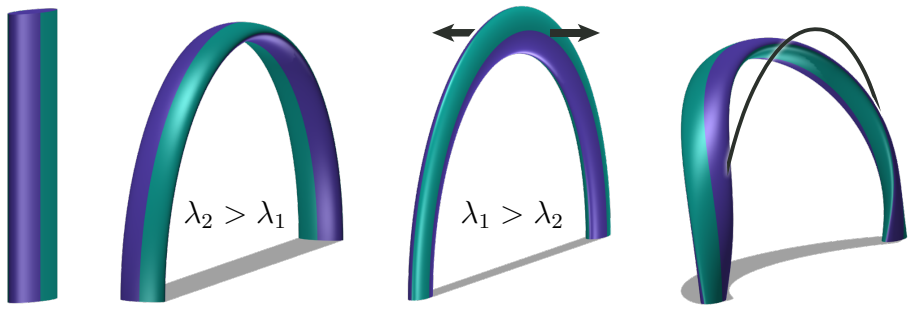


Figure 5.12: **Cross Sections & Stability.** *From left to right:* undeformed rod; stable equilibrium attained by aligning the major axis of the cross section with the bending axis; unstable equilibrium attained by doing the opposite; and the resulting buckled solution.

of buckling, it is a heuristic and not a formal stability guarantee. Indeed, for curves with many helical windings, a stable choice of I will likely not exist. If it is not intuitively clear whether a design is stable or not, it may be necessary to check for stability issues numerically prior to fabrication, or to verify the formal stability conditions for the solution [MRM98].

5.7 General Equilibrium Curves

To take advantage of the full design space offered by Kirchhoff rods, we need to bring twist into the equation. This goal requires a fundamental design decision: Does the burden to specify the twist fall on the designer, or is it done computationally? Both choices have a drawback: The designer may have a good idea of the curve shape they want, but may lack the intuition to know what twist will work to realize it. However, relying on a computational solution makes it more difficult to ensure that the twist falls into physically plausible bounds, as we will see.

For most of this section, we explore the latter choice, in which the designer only prescribes the deformed center line of the rod, and leaves the computation of twist and cross sections to the computer. This leads to the characterization of *general equilibrium curves*—curves in $\mathbb{R}^3$ that can be framed to yield a solution to Eq. 5.6, the equilibrium equation for Kirchhoff rods. Like in the parallel case, the characterization yields conditions that are linear in the unknowns and can thus be checked with a linear program. However, general equilibrium curves prove to be free of pointwise equality constraints, so the linear program is a useful tool for computational design and will work on manually drawn input curves, for example splines.

The challenge with this workflow is to find a solution that not only satisfies the equilibrium equation, but is also practical for fabrication. Two relevant criteria are that cross sections should not vary too much in size across the length, and that the twist should stay within reasonable bounds. We will show how these goals can be achieved without jeopardizing linearity.

At the end of this section, we also present a theoretical result about *framed equilibrium curves*, in which the curve and the moving frame are both prescribed. While we do not use this result for design applications, it provides a general insight into the structure of equilibrium states attainable within the theory of Kirchhoff rods.

5.7.1 Geometric Characterization

As a proper superset of parallel equilibrium curves, the conditions for general equilibrium curves must be strictly weaker than those in Proposition 7. Indeed, for Frénet curves, the characterization emerges simply by removing the tangent line condition, so only the binormal line condition remains.

We can view curves with continuous geometric curvature as elements of $C^2((0,1);\mathbb{R}^3)$ by relaxing the requirement of an arc-length parametrization. With the standard C^2-norm, Frénet curves are dense within the set of all regular curves. This implies that inflection points of curves in $\mathbb{R}^3$ are non-essential features that can be removed by a perturbation of arbitrarily small magnitude—this is in contrast to plane curves, where inflections are *not* removable.

We opt to prove the characterization only for Frénet curves, because the resulting theory suffices for the application explored in this work: to turn user-drawn spline curves into Kirchhoff rod equilibrium states.

Proposition 9. *Let $\gamma : (0,\ell) \to \mathbb{R}^3$ be an arc-length parametrized Frénet curve. Then, γ is an equilibrium curve if and only if there is a linear complex that contains none of the binormal lines of γ.*

Proof. "⇒": Assume there exist $c, \bar{c} \in \mathbb{R}^3$ and $K : (0,\ell) \to \mathcal{K}$ such that $FKk = c \times \gamma + \bar{c}$. We can write the normal component of this equation as $EIk_n = F_n^t(c \times \gamma + \bar{c})$. From $I \in S^2_{++}$ and the Frénet assumption $\kappa = \|k_n\| > 0$, it follows that

$$0 < \langle k_n, F_n^t(c \times \gamma + \bar{c})\rangle = \langle \omega_n, \bar{c}\rangle + \langle \gamma \times \omega_n, c\rangle \tag{5.16}$$

by the same computation as in Eq. 5.12. This shows that $(c, \bar{c})$ is a linear complex that does not contain any of the binormal lines, which have Plücker coordinates $(\omega_n, \gamma \times \omega_n)$.

"⇐": Assume there exists a linear complex $c, \bar{c} \in \mathbb{R}^3$ that contains no binormal line of γ. By continuity, we have $0 < \langle \omega_n, c \times \gamma + \bar{c}\rangle$, possibly after flipping the signs of c and $\bar{c}$. Choosing a parallel frame F adapted to γ yields $0 < \langle k_n, F_n^t(c \times \gamma + \bar{c})\rangle$, with k the curvature vector of F. Thus, there exists $I \in S^2_{++}$ such that $EIk_n = F_n^t(c \times \gamma + \bar{c})$.

Next, choose J such that $0 < J \leq 4\psi(I)$, and define

$$\tau_\beta = \frac{1}{\mu J}\langle \gamma', c \times \gamma + \bar{c}\rangle, \quad \text{and} \quad \beta(s) = \int_0^s \tau_\beta. \tag{5.17}$$

Then, the frame F_β defined by Eq. 5.1 has material curvatures $k_{\beta,n} = Q_\beta^t k_n$ and twist τ_β. Assembling the tangential and normal parts of $c \times \gamma + \bar{c}$ gives

$$EF_{\beta,n} Q_\beta^t I Q_\beta k_{\beta,n} + \mu J \tau_\beta \gamma' = c \times \gamma + \bar{c},$$

so F_β solves the equilibrium equation with $(Q_\beta^t I Q_\beta, J) \in \mathcal{K}$. □

In summary, every choice of $c, \bar{c} \in \mathbb{R}^3$ that satisfies

$$\langle \gamma' \times \gamma'', c \times \gamma + \bar{c}\rangle > 0 \tag{5.18}$$

enables us to pick cross sections that solve the equilibrium equation. Even if we limit the choice of (I, J) to elliptical cross sections, we have a one-dimensional family of ellipses to choose from at every point. This choice will influence both the twist and the ratio between the ellipse radii a and b. In the next section, we discuss how to compute c and $\bar{c}$ in a way that favors low twist and a reasonably small gap between a and b.

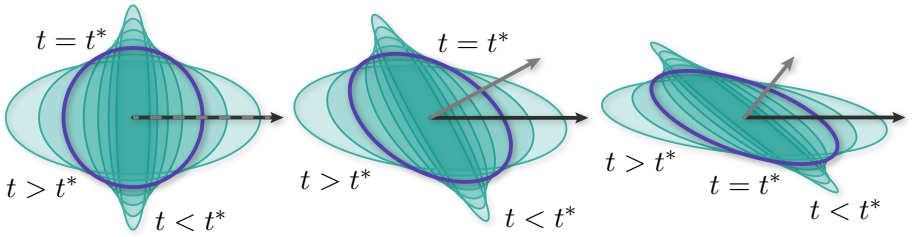


Figure 5.13: **Elliptical Families.** Family of elliptical cross sections that solve the equilibrium equation for fixed k_n (black) and $F_n^t(c \times \gamma + \bar{c})$ (gray). The angle between the two vectors increases *from left to right*. The most circular ellipse in each family ($t = t^*$) is marked purple.

5.7.2 Linear Program

To choose c and $\bar{c}$, we recall a result from Section 5.6 about parallel equilibrium curves: A non-plane Frénet curve has a parallel equilibrium frame if and only if we can find c and $\bar{c}$ that, in addition to Eq. 5.18, also satisfy $\langle \gamma', c \times \gamma + \bar{c} \rangle = 0$. As discussed in Section 5.6.3, this implies that k_n and $F_n^t(c \times \gamma + \bar{c})$ are parallel at every point, with $m = \kappa / \|c \times \gamma + \bar{c}\|$ the factor of proportionality. Thus, one solution to $EIk_n = F_n^t(c \times \gamma + \bar{c})$ is given by $I = \mathcal{I}_2/(Eq)$, with $\mathcal{I}_2$ the 2-by-2 identity matrix. This solution corresponds to a circular cross section, so $a = b$. This shows that parallel equilibrium curves satisfy both criteria for fabricability to perfection: they have zero twist, and allow us to choose a/b as close to unity as we like.

Unfortunately, the pointwise constraint $\langle \gamma', c \times \gamma + \bar{c} \rangle = 0$ cannot be satisfied exactly for most curves, but we can attempt to satisfy it approximately. In order to keep the problem linear, a natural choice is to minimize $\sup |\langle \gamma', c \times \gamma + \bar{c} \rangle|$. Using Eq. 5.17 (left), we can also interpret this objective mechanically as the minimization of the twist couple $\mu J \tau_\beta$. Next, we recall from Proposition 8 that parallel equilibrium curves also satisfy $\langle \gamma'', c \times \gamma + \bar{c} \rangle = 0$. Likewise, we can interpret this expression mechanically by computing

$$\langle \gamma'', c \times \gamma + \bar{c} \rangle = \langle \gamma', c \times \gamma + \bar{c} \rangle' = \mu (J \tau_\beta)',$$

so low values of $|\langle \gamma'', c \times \gamma + \bar{c} \rangle|$ correspond to a twist couple function with high C^1-regularity. We combine both objectives by minimizing $\sup |B^t (c \times \gamma + \bar{c})|_*$, with $B = (\gamma', \gamma''/\kappa)$ the orthonormal matrix having the tangent and principal normal as its columns, and

$$|(x, y)^t|_* = \max\{|x|, w_{\text{reg}} |y|\},$$

where $w_{\text{reg}} \geq 0$ is a user-controlled regularization weight. Finally, our linear program for determining c and $\bar{c}$ reads

$$\begin{aligned} &\text{minimize } R, \\ &\text{subject to } 1 \leq \langle \gamma' \times \gamma'', c \times \gamma + \bar{c} \rangle, \\ &\qquad -R \leq \langle \gamma', c \times \gamma + \bar{c} \rangle \leq R, && \text{(5.19a)} \\ &\qquad -R \leq w_{\text{reg}} \langle \gamma''/\kappa, c \times \gamma + \bar{c} \rangle \leq R, && \text{(5.19b)} \end{aligned}$$

where the constraints are enforced at a dense set of samples along the curve. This linear program in the variables $(R, c, \bar{c})$ can be solved near-instantaneously, so the user can interactively browse the family of solutions generated by varying the only parameter, w_{reg}.

5.7.3 Geometry Generation

Given c and $\bar{c}$, the next step is to choose a matrix $I \in S^2_{++}$ at every point that solves $EIk_n = F_n^t(c \times \gamma + \bar{c})$, where k_n and $F_n^t(c \times \gamma + \bar{c})$ are not necessarily parallel. The equation

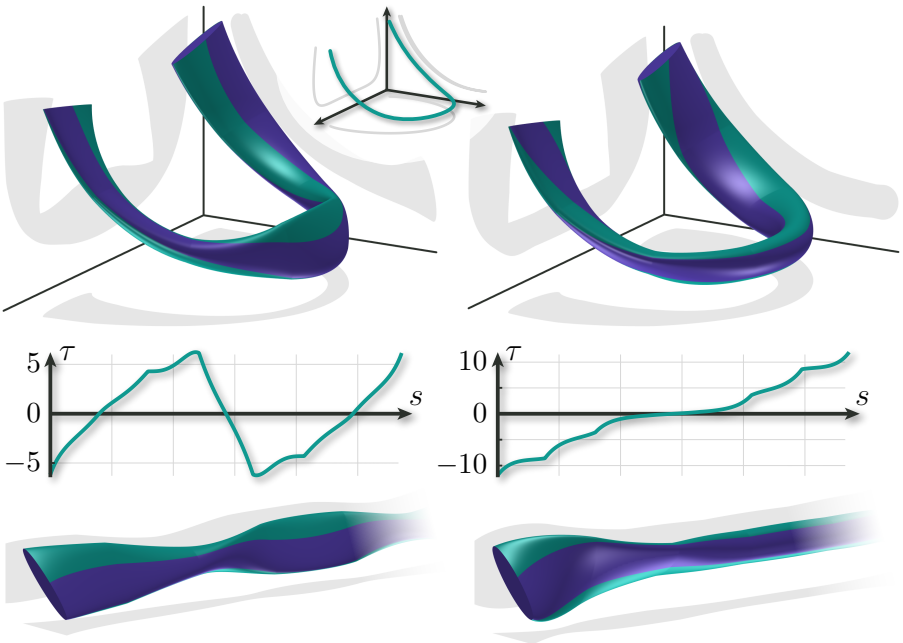


Figure 5.14: **Hook Curve.** Result of the geometry generation algorithm for general equilibrium curves used on a spline curve *(top center)* without *(left)* and with regularization *(right)*. *Top:* Deformed configuration. *Center:* Twist as a function of arc length. *Bottom:* Undeformed configuration.

constrains two out of three independent entries of S^2_{++}, which leaves a one-parametric family of solutions. We can compute an explicit representation of the form $I = I_1 + tI_2$, where I_1 and I_2 are symmetric positive semi-definite rank-1 matrices, and $t > 0$ is the free parameter.

Fig. 5.13 shows the family of solutions for different angles between k_n and $F^t_n(c \times \gamma + \bar{c})$. A canonical choice that exists in every family and benefits fabricability is the "most circular" ellipse, for which the ratio a/b is closest to unity. A symbolic computation shows that this solution can be found easily by choosing t as $t^* = \operatorname{tr} I_1 / \operatorname{tr} I_2$.

For $t \to 0$, the ellipse elongates orthogonally to $F^t_n(c \times \gamma + \bar{c})$, while for $t \to \infty$, it elongates along k_n. The latter property is useful for curves that are similar to parallel equilibrium curves, so the family of solutions looks like the one in Fig. 5.13 (left) at most points of the curve. Then, choosing $t > t^*$ instead of $t = t^*$ avoids stability issues for the reasons discussed in Section 5.6.3. After choosing I, we compute $J = 4\psi(I)$, τ_β and β via Eq. 5.17. Finally, we can compute $Q^t_\beta I Q_\beta$ at every point, which gives the elliptical cross section with the correct rotation.

5.7.4 Interactive Design

Fig. 5.14 shows the resulting geometry for a horseshoe-shaped input curve that bends out of plane. Even though it is not intuitively obvious what geometry and forcing mechanism will result in a curve like this, our algorithm finds a solution in which the out-of-plane deformation is induced by torque applied to the endpoints. The user interface provides control over w_{reg} in order to explore the trade-off between the objective in Eq. 5.19a and the regularization term in Eq. 5.19b. Prioritizing the former yields the solution on the left, with smaller overall twist, while increasing the regularization weight reduces the total variation, shown on the right.

To validate the solution, we perform a forward simulation of the Kirchhoff rod with the cross

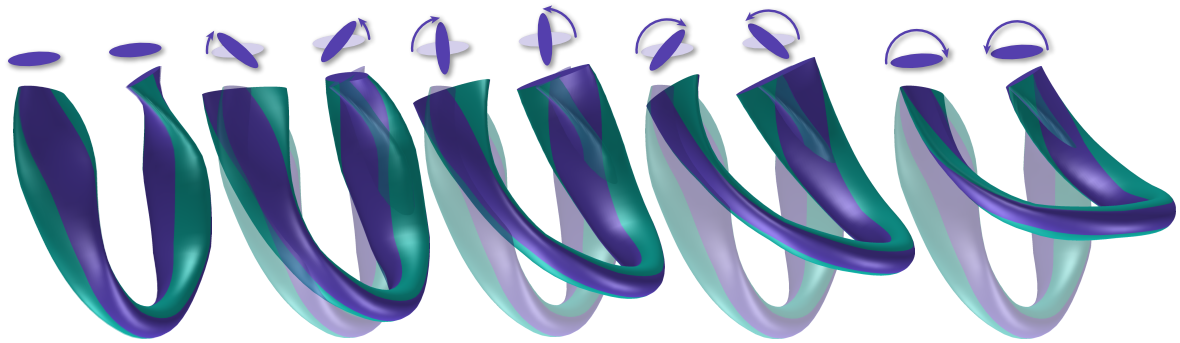


Figure 5.15: **Hook Curve Validation.** Forward simulation of a Kirchhoff rod generated by the computational design algorithm, to verify that the equilibrium state can be reached. The boundary conditions are applied gradually by twisting the endpoints.

sections computed by the computational design algorithm. The boundary conditions are applied by first bending the rod, and then twisting its endpoints to cause the out-of-plane deformation. Fig. 5.15 shows frames from the resulting animation, verifying that the desired equilibrium state can indeed by reached by continuous deployment.

5.7.5 Framed Equilibrium Curves

We conclude this section with a characterization of the set of all framed equilibrium curves. Here we consider the moving frame as part of the "input" instead of deriving it from the input curve. This results in a description of all equilibrium states that can be attained within the theory of Kirchhoff rods with zero natural curvature.

Looking back at Propositions 7 and 9, we find that the non-linear upper bound $J \leq 4\psi(I)$ present in the constitutive relation was not actually used in their proofs, so these statements hold independently of the existence of an upper bound. This is different in the following result, in which the upper bound imposes another inequality constraint on the design space. Surprisingly, the characterization remains linear in the unknowns c and $\bar{c}$ nevertheless and can thus be checked by a linear program.

Unlike zeros of the geometric curvature κ, zeros of τ are not removable by perturbations and need to be allowed to arrive at a useful characterization. To simplify the proof, we add the technical assumption that the zeros of τ are isolated, which is similar to assuming a finite number of inflection points in Theorem 6. Furthermore, we explicitly enforce coercivity and boundedness of the constitutive relation as a way of controlling possible singularities of $1/\tau$.

Proposition 10. *Let $\gamma \in C^2((0,\ell);\mathbb{R}^3)$ be an arc-length parametrized Frénet curve and $F : (0,\ell) \to SO(3)$ a moving frame adapted to γ such that all zeros of τ are isolated. Formally define*

$$v_1 = \frac{\gamma'}{\tau}, \quad v_2 = \frac{4\omega_n}{E\kappa^2} - \frac{\gamma'}{\mu\tau}.$$

Then, the following are equivalent:

1. *The framed curve (γ, F) is an equilibrium curve with a constitutive relation $(I, J) : (0,\ell) \to \mathcal{K}$ that is coercive and bounded.*[2]

[2]There exist $c, C > 0$ such that for all $s \in (0,\ell)$, it holds that $c \leq J(s) \leq C$ and $c \leq \lambda_1(s), \lambda_2(s) \leq C$ with $\lambda_1 \leq \lambda_2$ the eigenvalues of I.

2. *There exist* $c, \bar{c} \in \mathbb{R}^3$ *such that* $\langle v_i, c \times \gamma + \bar{c} \rangle > 0$ *for* $i = 1, 2$ *in the sense that the inequalities hold away from the zeros of* τ*, and the upper and lower limits are finite and positive as* $\tau \to 0$.

The proof of this characterization consists mostly of computations similar to the ones in Proposition 5 and arguments about the equivalence of the coercivity in (1) and the limits in (2). We carry out this proof in Appendix B.4.

5.8 Kirchhoff Rods under Load

The geometric characterizations in the preceding sections are subject to the limitation that we consider only the elastic energy of Kirchhoff rods, and neglect other factors such as the dead load and external loads. However, our computational design algorithms can be adapted to scenarios in which external loads have a small to moderate effect on the equilibrium state.

We will preface this section with the caveat that our proposed algorithm is, in contrast to the ones presented previously, a heuristic and comes without a formal guarantee for finding solutions to all feasible inputs. However, it terminates successfully for all reasonable inputs we have tried, including the examples shown in Section 5.9. We include the algorithm here it for its practical relevance, and as a starting point for future work on this subject.

5.8.1 Load Model

To model line loads (such as the dead load) and point loads (such as a weight hanging from a specific point) together, we use a *load distribution* $q(s) = p(s) + \sum_{i=1}^{n} \delta(s - s_i) f_i$. Here, $p : (0, \ell) \to \mathbb{R}^3$ models line loads (force per length) applied to γ, and $f_i \in \mathbb{R}^3, i = 1, \dots, n$, are concentrated forces applied at $\gamma(s_i)$, which are modeled as delta distributions. Integrating q on an interval $I \subset (0, \ell)$ gives the accumulated force applied to the rod on this interval.

The effect of this load distribution is captured by adding the potential $-\int_0^\ell \langle q, \gamma \rangle$ to the Lagrangian from Eq. 5.5. This modification leads to the equilibrium equation

$$(FKk)(s) = (c + Q(s)) \times \gamma(s) + (\bar{c} + M(s)), \tag{5.20}$$

where $Q(s) := \int_0^s q$ is the accumulated force, and $M(s) := \int_0^s \gamma \times q$ the accumulated moment.

5.8.2 Design Algorithm

The feasibility condition from Eq. 5.18 can easily adapted to a given load case by substituting c and $\bar{c}$ with the expressions appearing in Eq. 5.20, which yields

$$\langle \gamma' \times \gamma'', (c + Q) \times \gamma + (\bar{c} + M) \rangle > 0.$$

If the load is thought to be fixed, we can check for the existence of $c, \bar{c} \in \mathbb{R}^3$ satisfying this condition using a linear program as usual.

However, if we consider the dead load of a rod, the rod geometry and the load are coupled, because the cross sections determine the weight per unit arc-length. More precisely, the choice of $(c, \bar{c})$ determines the cross sections, which in turn determine the load, and thus M and Q, which appear in the equilibrium equation. This dependence of M and Q on $(c, \bar{c})$ is non-linear, so we can no longer check for feasibility with a linear program.

Geometry and Dead Load In preparation of our computational design algorithm under dead load, we discuss the relationship between rod geometry and dead load in more detail. The geometry is encoded in the bending stiffness $I(s)$ via Eq. 5.7, and the dead load is encoded as a line load $p(s) = A(s)\rho g$, where $A(s)$ is the cross-sectional area at $s \in (0, \ell)$, $\rho > 0$ is the material density, and $g \in \mathbb{R}^3$ is the gravitational acceleration.

Next, we define two maps: one from I onto p, and one from p onto I. The first map, $p(I)$, computes the dead load for a (fixed) geometry via $p(s) = 2\sqrt{\pi}(\det I(s))^{1/4}\rho g$. This holds because $A(s) = 2\sqrt{\pi}(\det I(s))^{1/4} = \pi a(s)b(s)$ is the area of an ellipse with radii $a(s), b(s)$. The second map, $I(p)$, computes the geometry that equilibrates a (fixed) load p. This map is defined by the steps in Section 5.7.3, with the difference that c is replaced by $c + Q$, and $\bar{c}$ by $\bar{c} + M$.

Note that the maps $p(I)$ and $I(p)$ are not generally inverses of each other: We only have $p^* = p(I(p^*))$ for some $p^* : (0, \ell) \to \mathbb{R}^3$ if γ is an equilibrium curve for the geometry $I(p^*)$ under its own dead load. Thus, solving the inverse design problem under dead load is equivalent to finding a fixed point of $p \circ I$.

Algorithm We propose to first fix $(c, \bar{c})$ heuristically (Step 1), and then apply a fixed-point iteration procedure (Step 2):

Step 1. We compute the load-free solution to Eq. 5.19, which yields constants c_0 and $\bar{c}_0$ and a first guess of the rod geometry, encoded by the bending stiffness $I_0(s)$. We compute the corresponding dead load $p_0 = p(I_0)$, and the accumulated force Q_0 and moment M_0. Our heuristic choice for the constants is $c := c_0 - \frac{1}{2}(\inf Q_0 + \sup Q_0)$ and $\bar{c} := \bar{c}_0 - \frac{1}{2}(\inf M_0 + \sup M_0)$. The rationale behind this is that the functions $c + M_0(s)$ and $\bar{c} + Q_0(s)$—which replace c_0 and $\bar{c}_0$—will be as close as possible to c_0 and $\bar{c}_0$, in the uniform norm.

Step 2. To find a fixed point of $p \circ I$, we iterate $p_{i+1} = p(I(p_i))$, until $\sup |p_{i+1} - p_i| < \varepsilon$, where we set $\varepsilon = 10^{-10}$ in our examples. We cannot offer a formal proof of convergence, but see experimentally that convergence is at least linear, and our examples terminate after less than ten iterations, which we show in Section 5.9.2.

5.9 Results

We implemented the computational design algorithms for parallel equilibrium curves (Section 5.6) and general equilibrium curves (Section 5.7) in a software tool that allows users to interactively design Kirchhoff rods. Edits made by the user trigger the computational design algorithms to re-run, and display the resulting geometry near-instantaneously to enable fast prototyping.

Once a design has been finalized, the resulting geometry can be exported for fabrication as parametric CAD geometry. This allows our tool to be integrated seamlessly into a CAD workflow, for example to create molds for fabrication, or design a support structure on which the Kirchhoff rods can be mounted.

Below we show objects designed with our algorithms and fabricated by casting silicone in 3d-printed two-part molds. The first example demonstrates the fabrication method and the design space of parallel equilibrium curves; subsequent examples use the algorithm for general equilibrium curves (with and without dead load and external loads), and show applications in soft robotics and design with active bending.

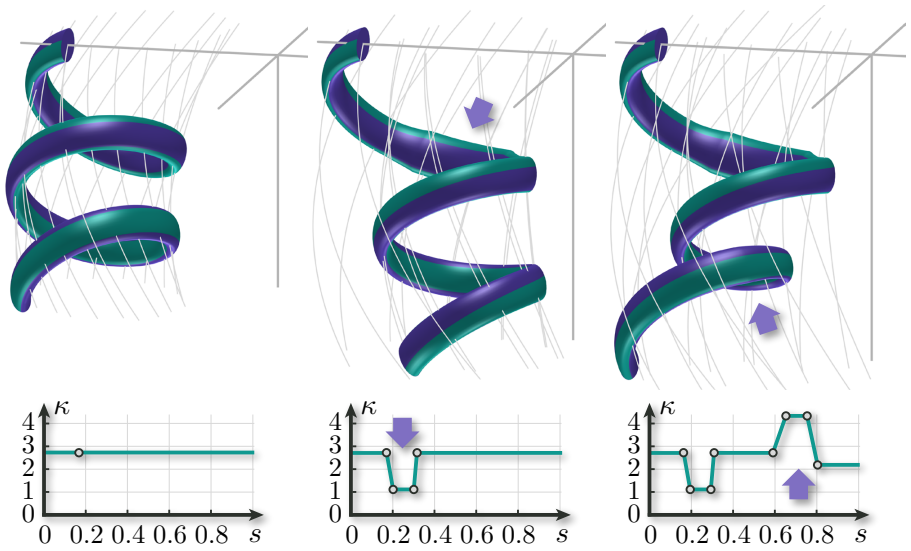


Figure 5.16: **Parallel Curve Design.** *Top:* In-app preview of the deformed rod geometry. *Bottom:* Editable geometric curvature. *From left to right:* Starting from a constant-curvature rod, the user adds more samples to the curvature graph to locally straighten or tighten the windings of the double helix. The curves from the helical motion (*gray*) serve as a visual guide.

5.9.1 Parallel Equilibrium Curve & Fabrication

Rod Design To design a Kirchhoff rod within the constrained space of parallel equilibrium curves (Section 5.6), the user is given control over the geometric curvature function κ and the three scalar parameters discussed in Section 5.6.2. Together, these quantities uniquely determine a parallel equilibrium curve up to rigid motion.

Initially, κ is set to be constant, which generates a segment of a double helix, but the user can modify κ by adding and dragging sample points in order to bend or straighten the curve in certain locations. Fig. 5.16 shows three snapshots from a design session, in which the user progressively edits a parallel equilibrium curve. The preview of the rod geometry that realizes this curve is updated in real-time as the control points are being dragged.

Mold Design Once a design is finished, it can be exported as a FeatureScript for the CAD system Onshape to generate solid parts of the undeformed and deformed rod geometries. The undeformed rod serves as a starting point for designing a 3d-printable mold, which is used for silicone casting during fabrication. To simplify this, our app also exports a *parting surface* that splits the mold into two parts, each guaranteed to be a height field.

The deformed rod is used to design a support structure, with sockets that enforce the kinematic boundary conditions at both endpoints of the rod. The automatically generated CAD geometry as well as the manually designed mold and support structure are shown in Fig. 5.17.

Fabrication We 3d-print the two-part mold from PLA on an Ultimaker S5, close it, and seal the seams with gaffer tape. For casting, the mold is placed vertically, which is important to prevent the formation of air bubbles. Next, we use a syringe to inject liquid silicone (SmoothSil 945) through the injection hole located near the bottom of the mold. The silicone rises through the cavity until it reaches the air vent at the top, at which point we seal the injection hole with a rubber plug.

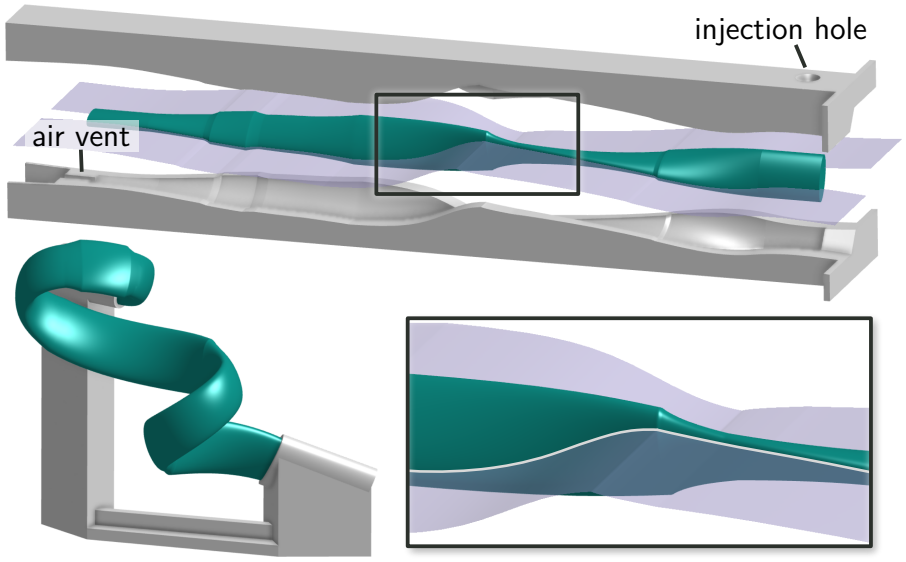


Figure 5.17: **Parallel Curve Mold.** *Top and bottom right:* Undeformed rod geometry (green) with parting surface (purple) and two-part mold (gray). *Bottom left:* Rendering of deformed rod (green) with support structure (gray).

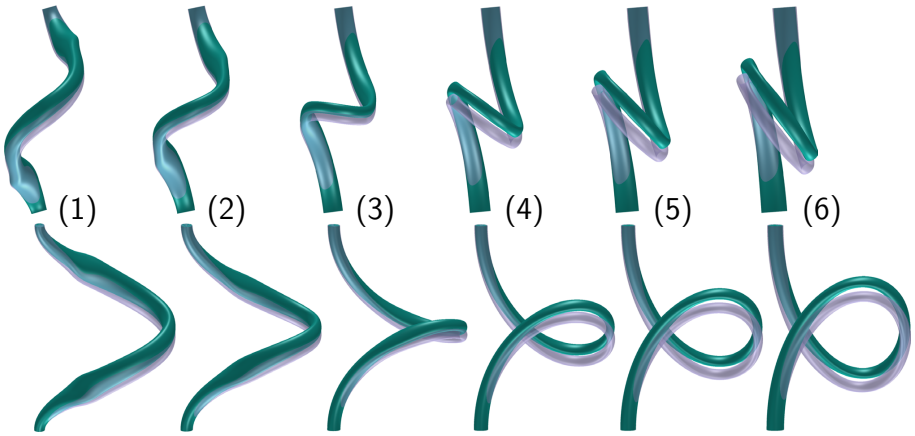


Figure 5.18: **Loop Array Deformed.** Front view (*top*) and side view (*bottom*) of rods from the loop array example. The final result, taking into account gravity for inverse design, is shown in green; the result of neglecting gravity during design, but forward-simulating with gravity, in purple.

After a 16-hour curing period, the rod is ready to be unmolded and mounted in the 3d-printed support structure, as shown in the photograph in Fig. 5.22. The helical windings, small thickness of the rod, and material chosen for this example all contribute to a low overall stiffness, which makes the rod sag visibly under gravity, compared to the target design. This illustrates the relevance of accounting for gravity in inverse design even on this scale—which we do in the following examples.

5.9.2 Loop Array

The addition of twist opens up the design space of Kirchhoff rods and gives more creative freedom to the designer by enabling direct control via spline editing tools or specifying curves analytically. These curves serve as input to the general equilibrium curve algorithm discussed in Section 5.7, and can be post-processed by the algorithm in Section 5.8 to account for the

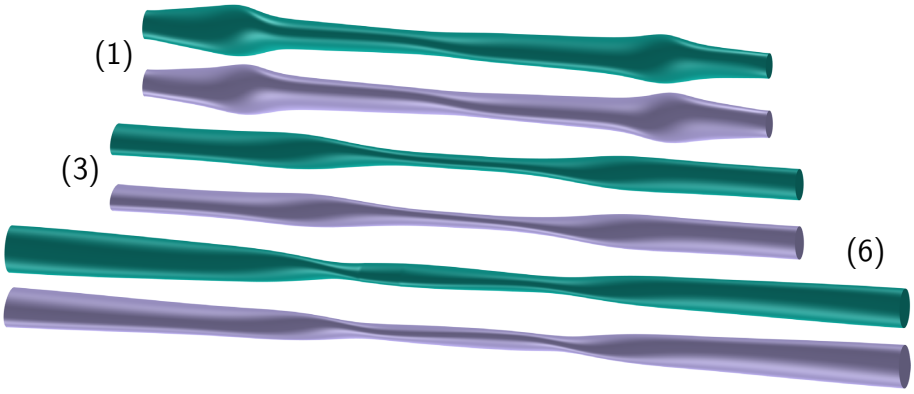
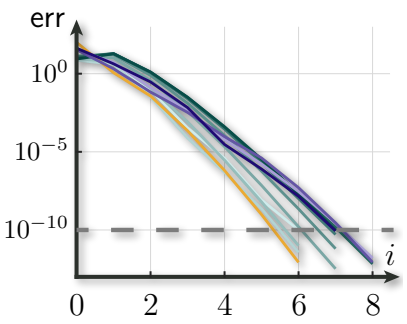


Figure 5.19: **Loop Array Undeformed.** Undeformed geometry of rods from the loop array, before (purple) and after (green) applying the dead load fixed-point iteration.

dead load.

In this example, we design and fabricate an array of six curves from a smooth family, shown in Fig. 5.18. The rods in purple are the result of solving the linear program in Eq. 5.19, which neglects gravity, and then forward-simulating the geometry *with* gravity. This makes the rods sag noticeably under their own weight, and prevents a faithful reproduction of the input curve.

Dead Load To improve reproduction of the input curve under gravity, we apply the dead load optimization algorithm from Section 5.8.2. The inset shows the fixed-point error $\sup |p_{i+1} - p_i|$ as a function of the iteration count i on a log scale, which provides numerical evidence that convergence is at least linear. Every graph in green corresponds to a rod from this example, while the ones in yellow and purple correspond to the rods from Sections 5.9.3 and 5.9.5, respectively.



The rods shown in green in Fig. 5.18 are the result of our optimization and reproduce the input curves perfectly (up to numerical error). The undeformed rod geometries before and after dead load optimization, as shown in Fig. 5.19, are visually very similar. This is because the geometry shown in purple serves as the initial value for the fixed-point iteration, which naturally converges to an attractive fixed point (green) close to it. Nevertheless, this small change suffices to counteract the effect of gravity. Renderings and photographs of the result are shown in Figs. 5.1 and 5.22 for direct comparison.

Performance Solving the linear program from Eq. 5.19 for one of the input curves (with 400 sample points) takes about 5 ms, and computing the rod geometry according to Section 5.7.3 an additional 6 ms. The dead load fixed-point iteration takes between 6 and 8 iterations to converge, and each iteration takes about 6 ms. Therefore, the total computation time per curve is about 60 ms, fast enough to show geometry changes due to user edits in real time.

5.9.3 Fixture Design

A popular application of bending-active materials is the design of structures that take their final shape only under the effect of a weight, such as a lampshade hanging from it, or a person sitting on it. Our system supports the design of objects like this by using the load distribution

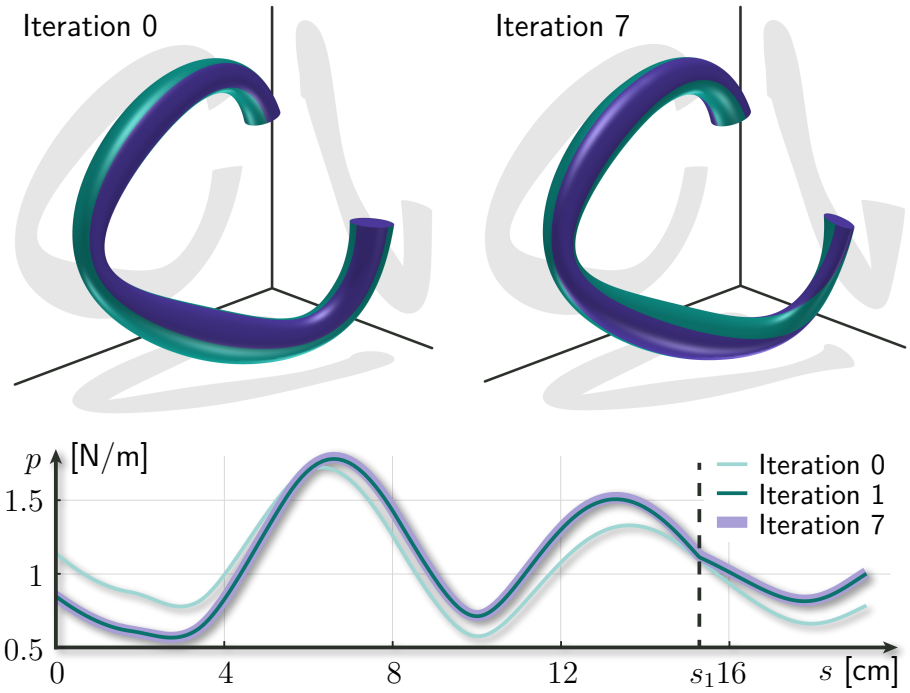


Figure 5.20: **Load Optimization.** *Top:* Deformed geometry before *(left)* and after *(right)* accounting for external load and dead load. *Bottom:* Vertical component of the dead load before and after fixed-point iteration.

described in Section 5.8, which models external line loads and point loads in addition to the dead load.

We demonstrate this feature by designing a fixture that is in equilibrium under its own weight plus a weight hanging from a specific point. The input curve is designed manually by manipulating the control points of a quartic B-spline curve. As usual, the user sees immediately after each edit how the geometry of the rod and the twist of the equilibrium state were affected by the change.

Load Optimization The external weight hanging from the rod is modeled as a point load $f_1 \in \mathbb{R}^3$ at a curve point $\gamma(s_1)$ as described in Section 5.8.1. During the fixed-point procedure, the load distribution $q(s) = p(s) + \delta(s - s_1)f_1$ takes into account both the dead load, which is updated in every iteration, and the external weight, which remains constant.

Fig. 5.20 shows the evolution of the dead load i during the fixed-point procedure. After a single iteration, the solution is converged enough for the graph of p to remain visually unchanged afterwards, and after seven iterations, the pointwise change is below $\varepsilon = 10^{-10}$. As seen in the top-right part of Fig. 5.20, the algorithm adds twist near the bottom endpoint. This extra twist has the effect of lifting up the hook enough to counter the force introduced by the external weight. The inset shows the final geometry, forward-simulated without the external weight (purple), in which case there is a large deviation from the target curve, and with the external weight (green), in which case the target curve is matched precisely. Fig. 4.24 offers a side-by-side comparison between photographs of the physical model and renderings of the target shape, showing a good agreement.

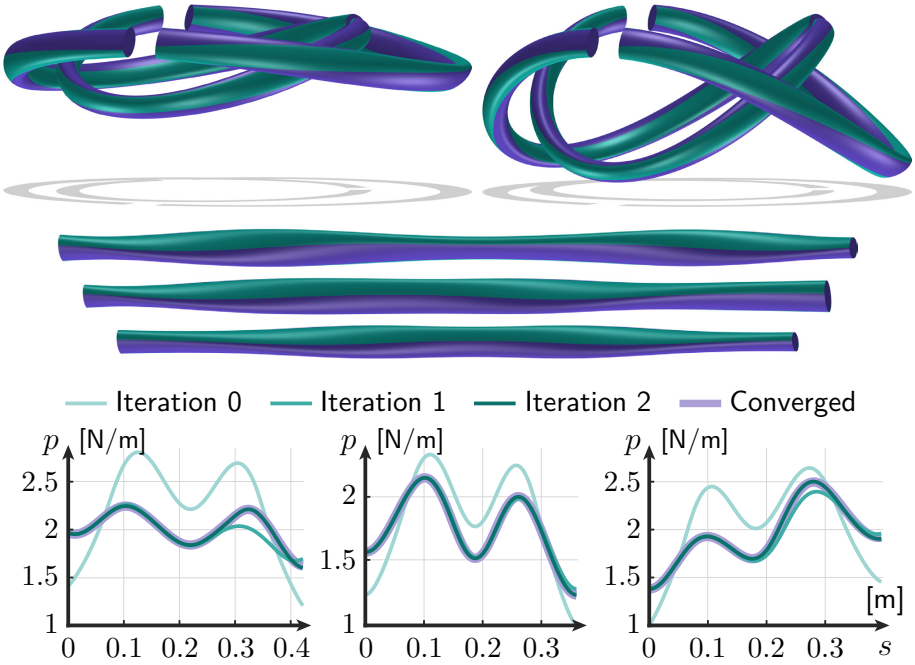


Figure 5.21: **Light Sculpture Design.** *Top:* Predicted deformed configuration of light sculpture if gravity is neglected *(left)* or accounted for *(right)*. *Center:* Undeformed configuration of all segments. *Bottom:* Iterations of the dead-load optimization for all three segments.

5.9.4 Soft Robotics

Elastomers like silicone are used in soft-robotics applications for the design of soft grippers or tools for minimally invasive surgery. Most of the soft-robotic mechanisms currently in use are actuated pneumatically or by cables [RDM19]. Both actuation systems add considerable complexity to the compliant part of the mechanism, through a sequence of air chambers or a network of cables around or inside the part.

With our algorithm, we can design simple compliant mechanisms that are actuated by twisting the endpoints of a rod. This does not add any complexity to the compliant part itself, because one only needs to add twisting joints to the fixture. The actuation itself can be performed by motors in the fixture, to follow a prespecified trajectory, or manually, if human fine-motor skills are required. Either way, this shifts the complexity away from the compliant part of the mechanism to an outside controller.

Lifting Tool We show a macro-scale version of a tool that can be used to lift objects out of an inaccessible location. The deformed configuration of the tool is given by the geometry shown in Fig. 5.14 *(right)*, where the out-of-plane deformation of the horse shoe is caused by twist applied to the endpoints. We 3d-print a mechanical fixture that connects the endpoints to rigid bars that a human user can turn to change the twisting angle from a distance, and to control the amount of out-of-plane bending.

The sequence of photographs in Fig. 5.22 shows a usage scenario, in which the tool is inserted into a tunnel with a sudden drop at the end. Using the twisting actuation, the flexible part of the tool bends downwards in order to grab a box and slide it up along the wall of the protrusion, so it can be pulled back through the tunnel.

5.9.5 Free-form Light Sculpture

We show an application to interior design, inspired by the *Freeform Light Sculpture* series of New York artist John Procario[3]. Our design mimics the organic wooden shapes of the original sculptures with black rubber beams that are made to follow a three-dimensional curved target shape by taking advantage of elastic and gravitational forces.

For this example, we forgo spline curves in favor of specifying the target design with a mathematical expression for a closed curve

$$\gamma(t) = \begin{pmatrix} r_1 \cos t + r_2 \cos(t/2 + p) \\ r_1 \sin t + r_2 \sin(t/2 + p) \\ a \cos(3t/2) \end{pmatrix},$$

with $t \in (0, 4\pi)$, and $r_1 = 1$, $r_2 = 1/4$, $p = 9/20$, and $a = 2/5$. We split the curve into three segments $(0, 4\pi/3)$, $(4\pi/3, 8\pi/3)$, $(8\pi/3, 4\pi)$, and compute separately for each the geometry of a Kirchhoff rod, taking into account the dead load as per Section 5.8.

Fig. 5.21 illustrates that gravity takes a central role in shaping the final deformed shape of this model. If we forward-simulate the undeformed geometry without gravity *(top-left)*, the resulting shape is very far from the target, while we have perfect agreement if gravity is accounted for *(top-right)*. The figure also shows the iterations of the dead-load optimization for each segment: After two iterations, the solutions are converged almost completely.

We manufactured the model using SmoothSil 960 silicone, with black pigments added for coloring. The design features a small indentation running along the inside of the rod, in which we place an electroluminescent wire. Figs. 5.1 and 5.22 show a rendering of the target design, and photos of the physical model with and without external lighting from a similar perspective, to allow for a visual comparison.

5.10 Discussion

In this work, we characterize the design space of Kirchhoff rods with spatially-varying cross sections and vanishing natural curvature. This geometric characterization gives rise to computational design algorithms that translate a curve in three-dimensional space to a Kirchhoff rod that attains this curve at equilibrium, given appropriate boundary conditions. We also discuss an extension that takes into account the effect of gravity, in order to enable applications on a larger scale.

A current limitation of our algorithm is that stability of the target equilibrium state is only enforced heuristically, and verified after the design stage through numerical means. However, stability cannot be rigorously guaranteed by our computational design algorithm. One could approach this issue by combining the adjoint method for stability optimization from Section 4.5 with the constrained Jacobi equation for Kirchhoff rods [MRM98]. Related to this problem is stability during deployment, such as the twisting actuation of our lifting tool: We can verify numerically that the trajectory of the Kirchhoff rod during actuation is stable, but we cannot automatically find a stable trajectory, or optimize the rod to make a certain trajectory stable.

Allowing arbitrary elliptical cross sections for rods has the disadvantage that one needs to use molding or 3-axis CNC milling for fabrication. It would therefore be of practical relevance

[3] `http://www.johnprocario.com/`

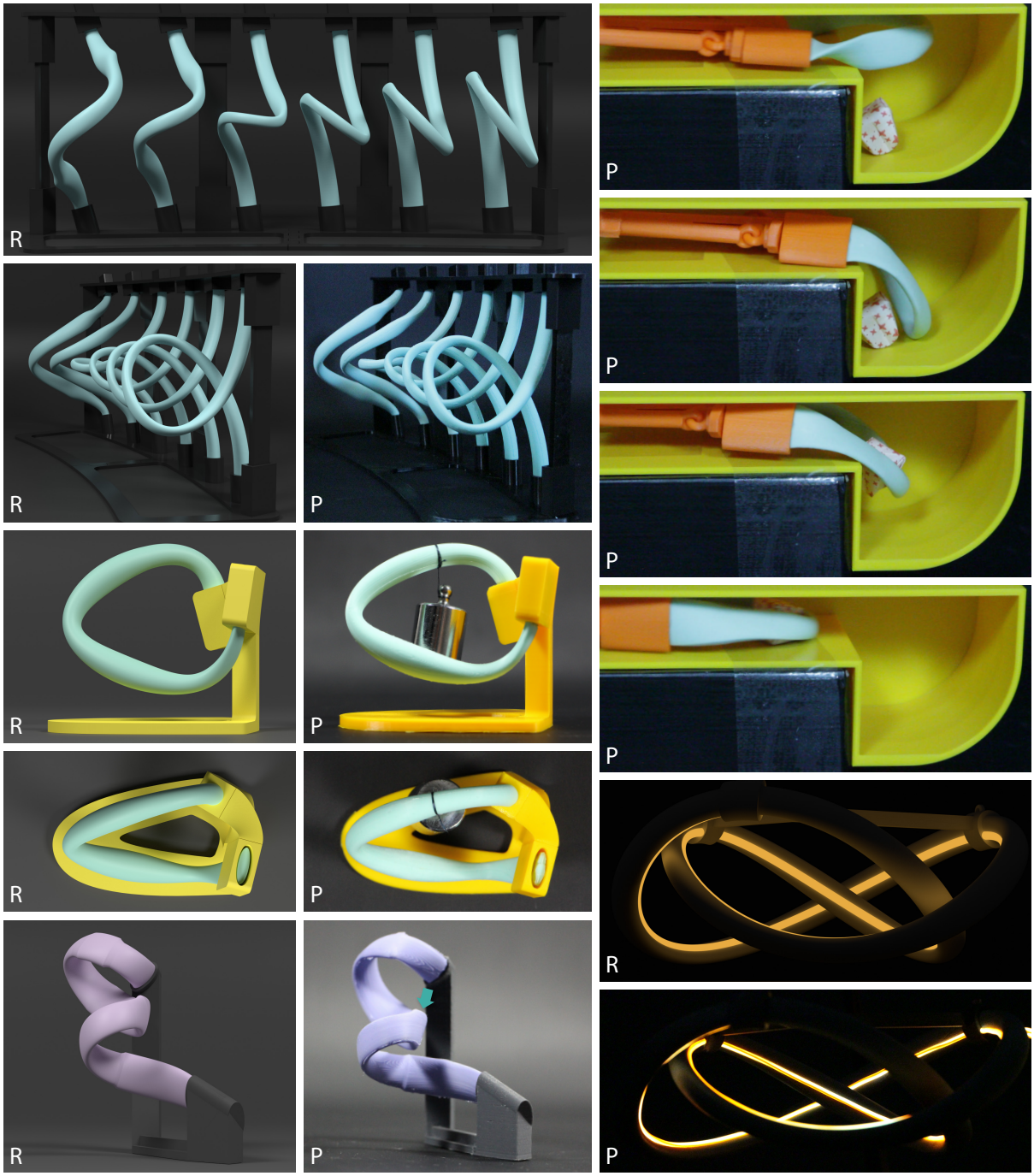


Figure 5.22: **Results.** Renderings (R) and photographs (P) of our models. *Cf.* renderings of loop array and light sculpture with photos in Fig. 5.1.

to study the design space of rods that can be produced with simpler means, e.g., laser cutting or water jetting rods from a sheet material. While this is unlikely to yield a geometric characterization as simple as the one studied in this paper, it may enable computational design algorithms with a wider range of applications.

CHAPTER 6

X-CAD: Optimizing CAD Models with Extended Finite Elements

6.1 Introduction

Ever since Ivan Sutherland laid the foundation of modern Computer-Aided Design (CAD) with his revolutionary computer program Sketchpad [Sut63], CAD systems have become a core pillar of innovation. In combination with simulation, they have empowered us to design architectural masterpieces like the Sidney Opera House, or fuel-efficient airplanes like the Airbus A350 XWB or the Boeing 787-9 Dreamliner. Yet, it remains onerous to treat CAD model parameters as design variables in optimizations.

In modern CAD systems, a *boundary* representation (B-rep), predominantly composed of Non-Uniform Rational Basis Spline (NURBS) patches, is used to describe solid models. The success of B-rep is attributed to the many desirable properties of NURBS, enabling the precise representation of analytical and free-form shapes, and modeling operations such as extrusion, chamfering, or blending. While advantageous for manual design, strength-to-weight or rest shape optimization require the solution of a Partial Differential Equation (PDE) on the enclosed *volume*.

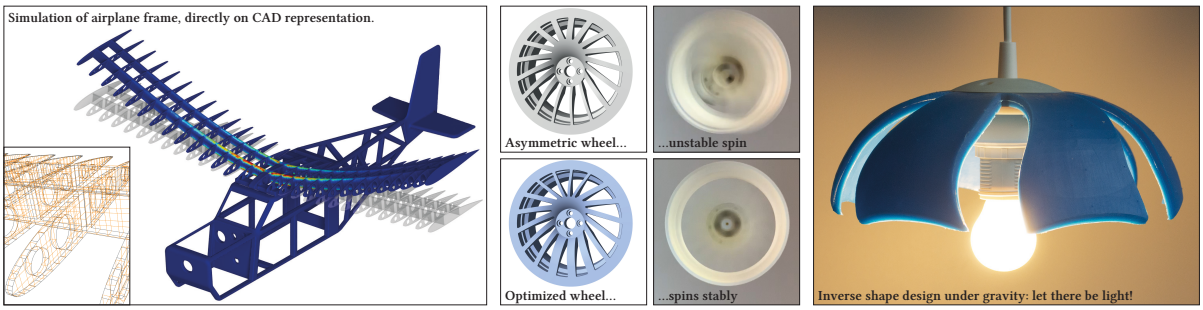


Figure 6.1: We present a differentiable deformable solid simulation (left) that enables shape optimization on CAD representations (middle, right) while preserving a model's manufacturability, function, and appearance. We demonstrate our optimization on a range of objectives including co-optimization of strength-to-weight ratio and mass distribution (middle), and rest shape optimization (right).

Although progress has been made in isogeometric analysis, where PDEs are solved on volumetric NURBS representations, the generation of volumetric NURBS for general B-rep input is highly challenging [CHB09]. Hence, it is still the de facto standard to solve PDEs on a volumetric *mesh* representation. However, because shape optimization requires a *differentiable* simulator, and even moderate changes to design variables demand repeated conversion and remeshing, the use of CAD in combination with optimization is limited.

In this paper, we propose a novel differentiable deformable solid simulation that enables generic shape optimization directly on CAD models. To dodge remeshing discontinuities and avoid a dependence of shape derivatives on the simulation mesh, we intersect the CAD model with a regular hexahedral grid that we keep *constant* throughout optimizations. While the resulting simulation mesh is not conformal, we enrich elements that are cut by the B-rep, representing the solid-void boundary *explicitly*. To accurately integrate over the subvolumes of the resulting extended finite elements, we adopt and extend a recent quadrature scheme [MKO13].

CAD models are often tailored for fabrication using a particular manufacturing technology. For example, if we target casting or injection molding, a model has to be undercut-free and observe a minimal draft angle constraint. To preserve a model's manufacturability, function, and appearance during optimization, we therefore analyze properties of neighboring NURBS patches, and provide the user with the option of defining constraints on the mapping of high-level shape to low-level patch parameters.

We demonstrate the efficacy, generality, and utility of our technique by minimizing common strength-to-weight, rest shape, and mass distribution objectives on a set of complex CAD models with a plethora of thin and sharp features. With a set of validation experiments, we further show that our XFEM simulation results are in excellent agreement with simulations performed on a conformal mesh with standard FEM, for both linear elasticity and hyperelasticity problems.

Succinctly, we propose and contribute

- a *differentiable* simulator that enables generic shape optimization on CAD models.
- an extension of a hierarchical quadrature scheme [MKO13] to accurately and reliably integrate subelement detail of varying shape and size.
- a change of basis for enriched elements, making it straightforward to turn a standard FEM into an efficient XFEM implementation.
- a differentiable projection of shape parameters onto a set of shape, function, and manufacturability constraints, and efficient shape derivatives of our hierarchical quadrature.

6.2 Overview

Before we delve into our technical contributions, we provide a high-level overview of how we simulate the elastic response of a CAD model, and optimize shape parameters with respect to objectives that depend on this response.

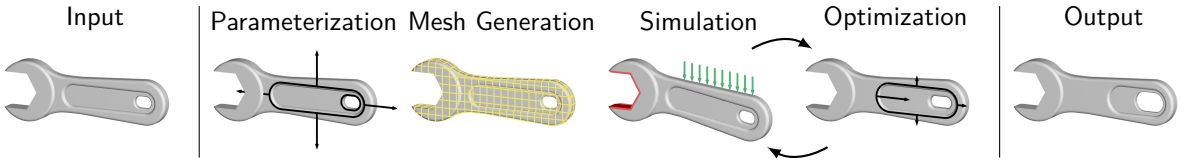


Figure 6.2: **Shape Optimization on CAD** Given a CAD model (Input), a user first selects shape parameters for optimization (Parameterization). Intersecting the model with a regular hexahedral mesh (Mesh Generation), we combine a novel integration scheme with XFEM to integrate finite element quantities with subelement precision (Simulation). Analytical shape derivatives of model-grid intersections, our integration scheme, and XFEM simulations (Optimization), enable function-, manufacturability-, and appearance-preserving shape optimization of CAD models (Output).

6.2.1 CAD Model Representation

The most general form of a CAD model that we consider is a closed NURBS mesh, i.e., a set of NURBS patches that form a C^0 surface. We assume that the modeler applied appropriate engineering judgment during initial design, so that the input fulfills *geometric* requirements for manufacturing (see Fig. 6.2, Input).

We rely on *projective* coordinates to represent NURBS patches, where points $[x, y, z]^T$ in Euclidean coordinates are represented with points $[wx, wy, wz, w]^T$ in projective space $\mathbb{P}^3$. We therefore assume a NURBS patch with control points $\mathbf{q}_{i,j} \in \mathbb{P}^3$ and polynomial basis functions $B_{i,j} : \mathbb{R}^2 \to \mathbb{R}$ to be a parametric mapping

$$\boldsymbol{\sigma} : \mathbb{R}^2 \to \mathbb{P}^3 \qquad \mathbf{u} \mapsto \sum_{i,j} B_{i,j}(\mathbf{u})\, \mathbf{q}_{i,j} \tag{6.1}$$

from uv-coordinates $\mathbf{u} = [u, v]^T$ to a point $\boldsymbol{\sigma}(\mathbf{u})$ in projective coordinates. In contrast to the rational form $\hat{\boldsymbol{\sigma}} : \mathbb{R}^2 \to \mathbb{R}^3$ in Euclidean space, this form is more convenient for simulation and optimization because $\boldsymbol{\sigma}$ is *polynomial*. We use this definition everywhere, and recover Euclidean coordinates by perspective division where necessary.

During optimizations, we seek to ensure that a model remains manufacturable, and that changes to shape parameters do not negatively impact its function or characteristic appearance. For instance, in CAD, it is commonplace to round off sharp edges and corners of models by introducing fillets. If we moved control points of patches in an uncontrolled manner, we could easily reintroduce sharp features between neighboring patches.

To prevent undesirable changes to the model, we put users in control, letting them define an implicit mapping from high-level shape parameters $\mathbf{p}$ to the set of m control points $\mathbf{q} \in \mathbb{R}^{4m}$ of the NURBS mesh

$$\mathbf{c}_{\text{para}}(\mathbf{p}, \mathbf{q}(\mathbf{p})) = 0. \tag{6.2}$$

During optimizations, we then enforce these constraints $\mathbf{c}_{\text{para}}$, keeping the number and topology of patches fixed. We defer a detailed discussion of our *parameterization* until Section 6.5.

A key benefit of our technique is that structural or related objectives are directly minimized on a CAD representation, and the optimized output (Fig. 6.2, Output) can be loaded into a modeling tool for further refinement, or to design the mold for manufacturing by casting or modeling.

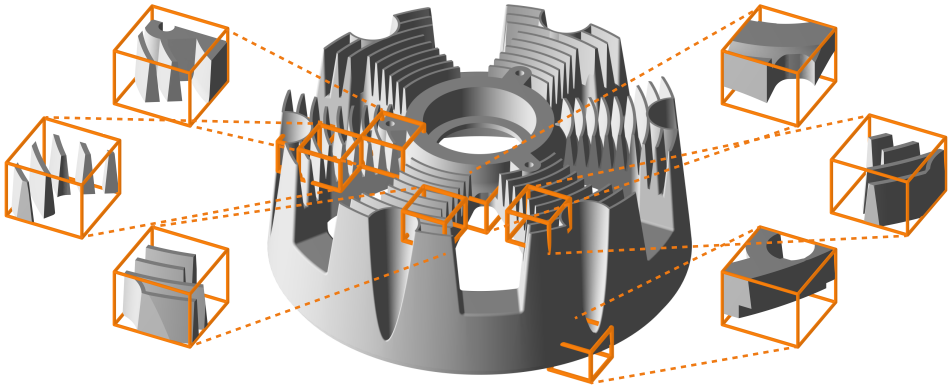


Figure 6.3: **Hexahedral Meshing** If we embed a CAD model in a regular hexahedral simulation mesh, the NURBS representation is cut into arbitrarily complex subvolumes near the boundary.

6.2.2 Deformable Solid Simulation on CAD

We target shape optimization of CAD representations where objectives depend on the elastic response of the material delimited by the boundary representation. To achieve this goal, our simulation has to be sufficiently *smooth* and *differentiable*. A standard conformal Finite Element (FE) discretization is ill-suited here because, if the shape of a model undergoes significant changes, remeshing is unavoidable. These uncontrolled topological changes lead to discontinuities, and therefore to a non-differentiable simulation.

To mitigate this problem, we propose to *embed* the CAD model in a regular hexahedral simulation mesh (Fig. 6.2, Mesh Generation). This mesh remains *constant* during optimization. To compute the elastic response $\mathbf{x} \in \mathbb{R}^{3n}$ of a CAD model (Fig. 6.2, Simulation), we seek to minimize the standard potential energy

$$E(\mathbf{x}) = E_{\text{int}}(\mathbf{x}) - E_{\text{ext}}(\mathbf{x}) \quad \text{with} \quad E_{\text{int}}(\mathbf{x}) = \int_V \Psi(\mathbf{x}, \mathbf{X})\, \mathrm{d}V, \tag{6.3}$$

where we integrate the material-dependent strain energy density Ψ [SB12] over points $\mathbf{X} \in \mathbb{R}^3$ in the undeformed volume V. The result of this minimization is a static equilibrium $E_{\mathbf{x}} = 0$ ($E_{\mathbf{x}}$ abbreviates the partial derivative $\frac{\partial E}{\partial \mathbf{x}}$) where the internal or elastic forces $E_{\text{int},\mathbf{x}}$ are in balance with external forces $E_{\text{ext},\mathbf{x}}$. However, in contrast to standard FEM, the volume V enclosed in the CAD model is the intersection of the B-rep with a regular hexahedral mesh, and elements on the boundary are cut into arbitrarily complex subvolumes as we illustrate in Fig. 6.3.

To represent the solid-void boundary in cut elements, *implicit* descriptions where signed distances to the boundary are discretized at mesh nodes, are common (see, e.g., [SZB18]). However, they fail to resolve subelement detail.

To loosen the coupling between mesh resolution and simulation precision, we therefore represent cuts in elements *explicitly* with an enrichment, and devise a quadrature scheme that integrates quantities such as the elastic energy E_{int} over complex subvolumes reliably and accurately. Although Koschier et al. [KBT17] recently addressed a related problem for simulations of detailed cuts by building on the quadrature scheme by Müller et al. [MKO13], a direct extension of their technique to our setting is not possible because integration over detailed curved domains with features smaller than a simulation element lead to large numerical errors as we epitomize in Fig. 6.4 bottom.

Based on this observation, we propose

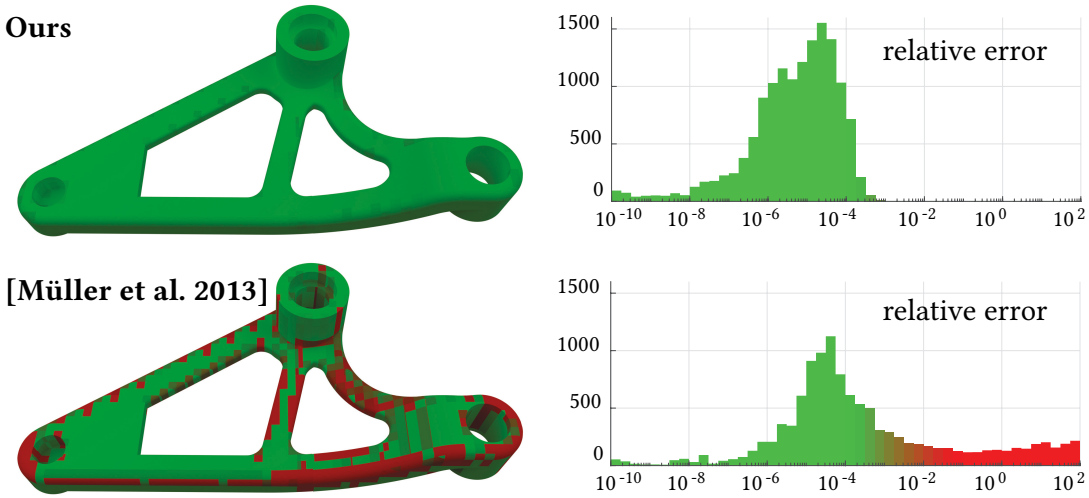


Figure 6.4: **Quadrature** For detailed subvolumes resulting from intersections of CAD models with regular simulation meshes, Müller et al.'s quadrature [MKO13, MKEKO17] (bottom) introduces large numerical errors while ours (top) is accurate. To generate the histograms on the right, we integrated the polynomial basis that is used for rule construction. To compute the relative error, we divided the absolute value of the difference of the exact and numerical integrals over the subvolume, by the absolute value of the exact integral over the element volume.

1. a modified set of quadrature rules that accurately handle integration over curved domains of varying shape and size, delimited by NURBS and planar patches (see Fig. 6.4 top).
2. a refinement of rule construction to significantly reduce the cost of evaluations of shape derivatives and updates to rules when shape parameters change.
3. a change of basis for enriched shape functions that makes it straightforward to turn standard FEM into efficient XFEM implementations.

We discuss our integration scheme in Section 6.3, and our XFEM formulation in Section 6.4.

6.2.3 Optimizing CAD Models

A first generic type of objective we seek to optimize integrates a function g that depends on the elastic response of the model over the volume enclosed by the B-rep

$$f(\mathbf{q}(\mathbf{p}), \mathbf{x}(\mathbf{p})) = \int_{V(\mathbf{q})} g(\mathbf{x}, \mathbf{X}) \, \mathrm{d}V. \tag{6.4}$$

Because the control points of the B-rep define the volume V, and changes to shape parameters translate to changes in control points, the rest shape of the model, hence also its elastic response, *implicitly* depend on the shape parameters $\mathbf{p}$.

Recent examples where this type of objective is used are traditional compliance optimization (see, e.g., [LHZ+18]) or the minimization of the potential of failure of structures [SZB18]. For compliance optimization, the strain energy density Ψ for linear elasticity is integrated over the rest volume, and for the minimization of the potential failure, a metric that measures the exponentiated distance of the Cauchy stress to the failure surface of a generic failure criterion, is integrated.

It is often desirable to minimize objectives that depend on the elastic response together with mass distribution objectives. For example, if we seek to optimize the strength-to-weight ratio of an asymmetric wheel design (compare with Figs. 6.1 and 6.15), the center of mass has to lie on the wheel's axis, and the major axis of the moment of inertia has to align with this axis [BWBSH14]. Otherwise, the model cannot fulfill its function.

To support the co-optimization of such combinations of objectives, we introduce a second type of objective that integrates standard functions over the volume delimited by the B-rep

$$f(\mathbf{q}(\mathbf{p})) = \int_{V(\mathbf{q})} g(\mathbf{X}) \, \mathrm{d}V. \tag{6.5}$$

Substituting an either constant or spatially-varying density $\rho(\mathbf{X})$ times a monomial $t \in \{1, X, Y, Z, XY, XZ, YZ, X^2, Y^2, Z^2\}$ for the integrand, we can compute a model's mass, its center of mass, and moment of inertia [BWBSH14]. For example, if we integrate the density (times the constant 1), we get the mass of a CAD model, and combined with our first type, we can formulate common strength-to-weight ratio optimizations.

A third type of objective we seek to support only depends on the elastic response of a model

$$f(\mathbf{x}(\mathbf{p})) = g(\mathbf{x}). \tag{6.6}$$

This type of objective enables, for example, inverse shape design [CZXZ14], where the rest shape of the model is optimized such that the deformed model matches a target shape under a predefined load as closely as possible.

Shape Optimization In our shape optimizations (Fig. 6.2, Optimization; Section 6.5), we then seek to minimize a single or a weighted combination of these objectives over the parameterized volume that a CAD model encloses

$$\min_{\mathbf{p}} f(\mathbf{p}, \mathbf{q}(\mathbf{p}), \mathbf{x}(\mathbf{p})) \quad \text{s.t.} \quad \begin{matrix} \mathbf{c}_{\text{para}}(\mathbf{p}, \mathbf{q}(\mathbf{p})) = 0 \\ E_{\mathbf{x}}(\mathbf{q}(\mathbf{p}), \mathbf{x}(\mathbf{p})) = 0 \end{matrix}, \tag{6.7}$$

enforcing first-optimality constraints on our parameterization and the elastic response. To prevent shape parameters from taking on values that would lead to non-manufacturable designs, we add an additional term to f that directly depends on $\mathbf{p}$ (e.g., penalizing the radii of two cylinders to prevent them from overlapping), hence the direct dependence of f on $\mathbf{p}$.

To enable shape optimization on CAD, we contribute

1. a *continuous* projection of shape parameters onto the constraint manifold spanned by the user-specified parameterization, guaranteeing well-posedness of the problem, and

2. a technique to *efficiently* compute derivatives of our hierarchical quadrature rules.

In Section 6.6, we demonstrate our technique on a wide range of examples including the compliance minimization of a motor housing, the inverse shape design of a lampshade, and the co-optimization of the strength-to-weight ratio and balance of an asymmetric wheel.

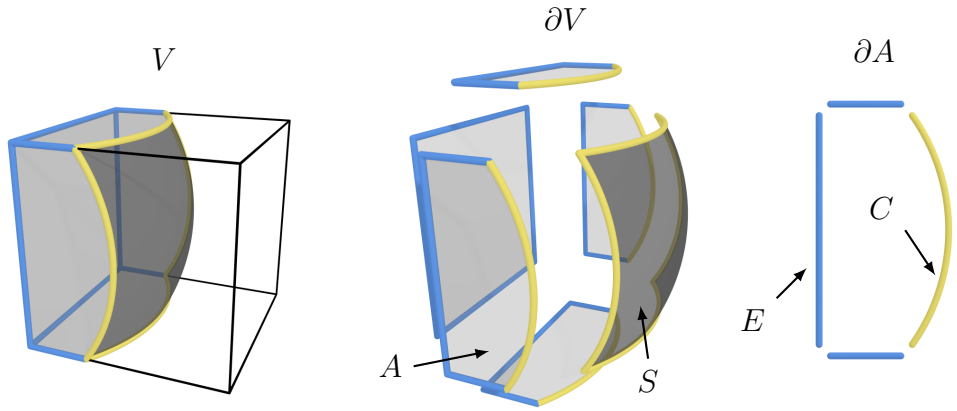


Figure 6.5: **Nesting of Integration Rules** To construct rules for integration over volumes V (left), we rely on rules for integration over the volume's boundary ∂V, decomposed into planar areas A (in light gray) and curved surfaces S (in dark gray). To generate rules for area and surface integrals (middle), we express integrals along the boundary ∂A with integrals along edges E and curve segments C (right).

6.3 Integrating over Subvolumes

To simulate a complex CAD model on a simulation mesh that is *not* conformal, we require quadrature rules for integration of functions over (1) subvolumes, which are part of the model interior (e.g., to accumulate elastic force density), and over (2) regions on the model surface (e.g., to aggregate surface traction).

In our setting, quadrature rules are not readily available because the integration domains are generated at runtime as intersections between arbitrary models and planes (see Fig. 6.3). Müller et al. [MKO13] proposed a moment-fitting technique to compute on-the-fly quadrature rules for domains with curved boundaries. Originally developed for integration of implicit representations and fluid simulation, we propose a variant that avoids failure cases when integrating over *explicitly* defined, detailed subvolumes (compare with Fig. 6.4), and—at the same time—increases computational efficiency of the technique.

In the following, we assume that the NURBS patches have been cut along edges and faces of the hexahedral mesh. To do so, we rely on *robust* algebraic curve tracing [BHLH88], and perturb grid planes to avoid corner cases. Refer to Fig. 6.5 for an example case of a patch-element intersection (left): Analogously to Müller et al. [MKO13], we build integration schemes in a hierarchical manner. We use edge and curve rules (right) to integrate over areas and surfaces, and area and surface rules (middle) to integrate over volumes (left).

Below, we use $g : \mathbb{R}^3 \to \mathbb{R}$ to denote a general function defined on the volumetric domain enclosed by the CAD model. Our goal is to construct quadrature rules that *exactly* integrate g, drawn from a function space spanned by a set of basis functions, over a domain D

$$\int_D g(\mathbf{X})\,\mathrm{d}D = \sum_j w_j g(\mathbf{X}_j). \tag{6.8}$$

The domain D is either one-dimensional, for integration along axis-aligned edges E (in blue in Fig. 6.5) or curves C (in yellow), two-dimensional, for integration over planar patches A (in light gray) or curved surfaces S (in dark gray), or three-dimensional, for integrals over volumetric domains V. Because we integrate over undeformed domains, we use capital $\mathbf{X}_j \in \mathbb{R}^3$ to refer to quadrature points corresponding to weights w_j.

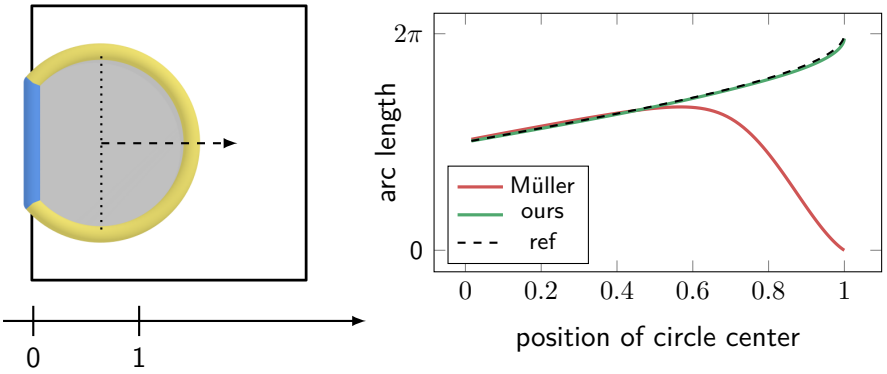
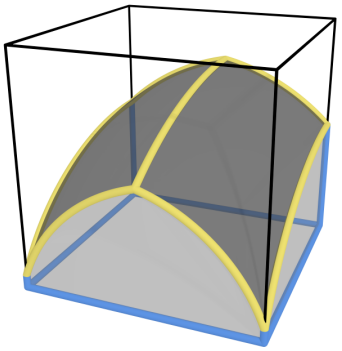


Figure 6.6: **Integrating Features** Designed for integrating over implicitly defined domains, Müller et al.'s method [MKO13] fails to integrate subelement detail as we demonstrate with a curve integral to compute the arc length of a cut circle (in yellow). While our integration accurately predicts the analytical arc length (green vs. dashed line), Müller et al.'s method is less accurate (see red line) the shorter the edge E (in blue) becomes.

6.3.1 Integrating along Edges and Curves

To integrate along axis-aligned edge segments E, we form one-dimensional integrals, for example $\int_{[a,b]} g(X,Y,Z)\,\mathrm{d}X$ for an integral along the X-axis. Like Müller et al. [MKO13], we perform a change of variables to map the interval $[a,b]$, delimited by the two segment endpoints, to the interval $[0,1]$, then apply a standard Gauss-Legendre rule.

Intersections of NURBS patches with hexahedral elements form *planar* curves that are embedded in planes parallel to one of the coordinate planes. To integrate along these curves, Müller et al. [MKO13] construct a divergence-free basis, and express curve integrals with sums of integrals along straight edge segments. However, as we illustrate in Fig. 6.6 with a circle example, their method loses accuracy as the chord length (in yellow) of the cut circle decreases. Another case that arises in our application domain is that patches intersect within elements, and form, in general, *spatial* curves as shown in the inset.



For accurate integration along curves C, we parameterize them with a mapping from $t \in [a,b]$ to spatial curve points $\mathbf{r}(t)$, then use a Gauss-Legendre rule for numerical integration of the transformed integrals $\int_{[a,b]} g\left(\mathbf{r}\left(t\right)\right) \left\|\mathbf{r}'\left(t\right)\right\| \mathrm{d}t$ where $\mathbf{r}'$ denotes the derivative of the mapping with respect to parameter t.

While we can expect intersection curves to be sufficiently smooth, we cannot, in general, extract analytical parameterizations from intersections of the B-rep with the hexahedral mesh. Hence, we represent them with sample points, and approximate its parametric form with a Lagrange interpolating polynomial. Note that the accuracy of the Gauss-Legendre integration is preserved for a polynomial interpolation of sufficiently high degree [AV93].

6.3.2 Integrating over Areas and Surfaces

To integrate over planar areas and surfaces, we make use of the first nesting of our hierarchical integration scheme. There are two cases to consider: (1) integrals over *planar areas* that lie

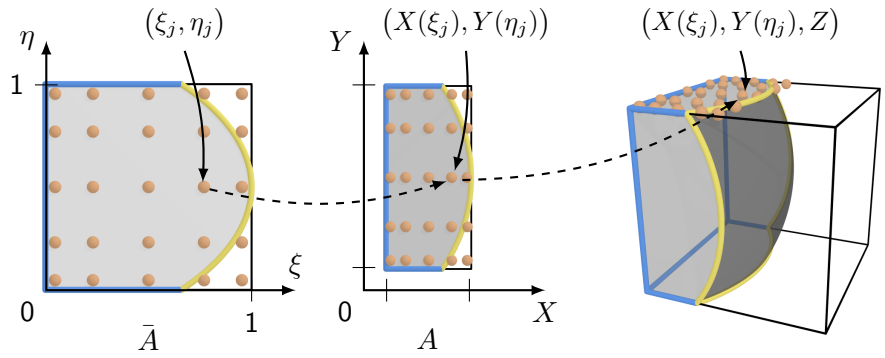


Figure 6.7: **Integrating over Areas** To integrate over planar domains A (right), we first compute an axis-aligned bounding box (middle), then transform the integral to the isoparametric domain $\bar{A}$ (left). In the domain $\bar{A}$, we use standard Gauss quadrature points (ξ_j, η_j) (in beige), and transform them back to spatial $(X(\xi_j), Y(\eta_j), Z)$ after rule construction. For surface integrals, we perform this transformation in the parameter domain.

in grid planes of the simulation mesh, and (2) integrals over *curved surfaces*, represented by NURBS patches.

Area Integrals For integration over planar domains A that are parallel to one of the coordinate planes, we use moment fitting analogously to Müller et al. [MKO13]. Moment fitting is similar to the construction of Newton-Cotes rules: given a set of predefined quadrature points, a system of equations is solved to compute corresponding quadrature weights such that a polynomial basis $\{p_1, \ldots, p_m\}$, spanned by a set of m functions, is integrated *exactly*. However, a crucial difference is that, due to the non-standard domain, basis functions cannot be integrated *analytically*, and this is the reason why a nesting is necessary.

There is a second important difference between Newton-Cotes and Müller et al.'s construction of rules: To avoid having to rederive rules whenever the domain $[a, b]$ changes, Newton-Cotes rules are constructed and tabulated for standard ranges (e.g., the range $[0, 1]$). For moment fitting, tabulation of rules is not possible due to the non-standard domains. However, transforming the area integrals such that the bounding boxes of the transformed domains coincide with the unit square $[0, 1]^2$ enables the use of a *single* system matrix for the construction of *all* our area and surface integrals. This bears far-reaching advantages: Whenever we make changes to shape parameters, rules have to be updated. If the system matrix does *not* depend on the parameters, the system can be prefactorized and rules updated more efficiently. In addition, we can avoid having to take derivatives of the inverse of a matrix, leading to a significant performance increase for evaluations of shape derivatives.

More formally, to integrate over an area A that lies in the XY-plane, we first compute an axis-aligned bounding box $[a, b] \times [c, d]$ (compare with Fig. 6.7 middle), then define a mapping between isoparametric variables ξ and η and the two spatial coordinates

$$\begin{pmatrix} X(\xi) \\ Y(\eta) \end{pmatrix} = \begin{pmatrix} b-a & \\ & d-c \end{pmatrix} \begin{pmatrix} \xi \\ \eta \end{pmatrix} + \begin{pmatrix} a \\ c \end{pmatrix}. \tag{6.9}$$

Because of the linearity of the mapping, its Jacobian is constant and the transformed integral reads

$$\int_{\bar{A}} g(X(\xi), Y(\eta), Z) \det \left(\frac{\partial(X, Y)}{\partial(\xi, \eta)} \right) \mathrm{d}\bar{A} \tag{6.10}$$

where $\bar{A}$ is the non-uniformly scaled domain, and the determinant of the Jacobian is set to the constant $(b-a)(d-c)$.

Moment Fitting For readers unfamiliar with moment fitting, we provide a brief technical overview here, pointing the interested reader to the original work [MKO13] for detail.

To compute the quadrature weights, we form the system

$$\underbrace{\begin{pmatrix} p_1(\xi_1,\eta_1) & \dots & p_1(\xi_n,\eta_n) \\ \vdots & \ddots & \vdots \\ p_m(\xi_1,\eta_1) & \dots & p_m(\xi_n,\eta_n) \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} \bar{w}_1 \\ \vdots \\ \bar{w}_n \end{pmatrix}}_{\mathbf{w}} = \underbrace{\begin{pmatrix} \int_{\bar{A}} p_1(\xi,\eta)\,\mathrm{d}\bar{A} \\ \vdots \\ \int_{\bar{A}} p_m(\xi,\eta)\,\mathrm{d}\bar{A} \end{pmatrix}}_{\mathbf{b}}$$

with constant matrix $\mathbf{A}$, evaluating the basis functions at Gauss quadrature points [MKO13]. For moment fitting to work, there need to be at least $n \geq m$ quadrature points, (ξ_j, η_j), forming an underdetermined system. To solve the system, we factorize the pseudo-inverse in the minimal-norm solution

$$\mathbf{w} = \mathbf{A}^T \left(\mathbf{A}\mathbf{A}^T\right)^{-1} \mathbf{b}. \tag{6.11}$$

While matrix $\mathbf{A}$ is independent of the integration domain, the right-hand side is not. To evaluate $\mathbf{b}$, we make use of the divergence theorem

$$\int_{\bar{A}} p_i(\xi,\eta)\,\mathrm{d}\bar{A} = \int_{\partial\bar{A}} \mathbf{n}(\xi,\eta) \cdot \mathbf{P}_i(\xi,\eta)\,\mathrm{d}\bar{s} \tag{6.12}$$

where $\mathbf{n}$ is the outward-facing normal at (ξ,η) and

$$\mathbf{P}_i(\xi,\eta) = \frac{1}{2} \begin{pmatrix} \int p_i(\xi,\eta)\,\mathrm{d}\xi \\ \int p_i(\xi,\eta)\,\mathrm{d}\eta \end{pmatrix} \tag{6.13}$$

the antiderivative, chosen such that $\nabla \cdot \mathbf{P}_i = p_i$. Note that the boundary of the domain $\partial\bar{A}$ consists of straight edge segments and planar curves (see Fig. 6.5 right), and we use rules developed in Section 6.3.1 to numerically integrate along them.

After construction, we transform the weights and quadrature points back to the original domain

$$w_j = (b-a)(d-c)\bar{w}_j \quad \text{and} \quad \mathbf{X}_j = [X(\xi_j), Y(\eta_j), Z]^T. \tag{6.14}$$

Surface Integrals For surface integrals, Müller et al. [MKO13] proceed analogously to the edge-curve case, and express surface integrals with a sum of integrals over planar areas. The resulting rules suffer from similar issues as the one-dimensional rules: if the planar areas become too small, the integration error increases uncontrollably.

For accurate integration over surfaces, we instead make use of the parametric form of NURBS patches, expressing them as *area* integrals in parameter space

$$\int_S g(\mathbf{X})\,\mathrm{d}S = \int_A g(\hat{\boldsymbol{\sigma}}(u,v))\,\|\hat{\boldsymbol{\sigma}}_u(u,v) \times \hat{\boldsymbol{\sigma}}_v(u,v)\|\,\mathrm{d}A. \tag{6.15}$$

Thus, integration weights are computed in uv-space, then transformed to physical coordiantes by multiplication with the area factor $\|\hat{\boldsymbol{\sigma}}_u \times \hat{\boldsymbol{\sigma}}_v\|$. Due to the non-linearity of this transformation, the resulting rule may not exactly integrate polynomials in physical coordinates. However, this error is mitigated by the smoothness of the area factor as we empirically show in Fig. 6.4.

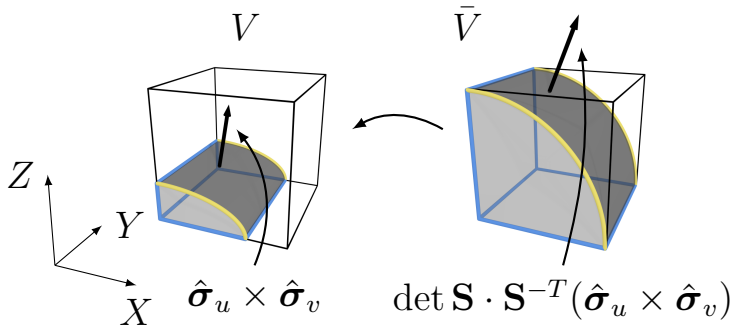


Figure 6.8: **Integrating over Volumes** To integrate over a volumetric domain V, we transform an axis-aligned bounding box of the volume to the unit cube $\bar{V}$. For consistency, normals $\hat{\boldsymbol{\sigma}}_u \times \hat{\boldsymbol{\sigma}}_v$ on curved surfaces need to be transformed before we can apply surface area rules. We use the linear transformation rule for cross products to do so.

6.3.3 Integrating over Volumes

To integrate over volumes, we proceed analogously to area integration (compare with Fig. 6.8): We first compute a bounding box $[a, b] \times [c, d] \times [e, f]$, and define a mapping to the unit cube. To evaluate the integrals of basis functions over the transformed domains, we use area and surface rules developed in the previous section, establishing a second and final layer of nesting.

An important detail is that the non-uniform scaling $\mathbf{S} = \text{diag}(b - a, d - c, f - e)$ has to be taken into account when we transform integrals over curved domains as we illustrate in Fig. 6.8. We use the rule for linear transformations of cross products to account for this scaling in our surface integrals

$$\int_A g(\mathbf{X}) \, \|\mathbf{S}\hat{\boldsymbol{\sigma}}_u \times \mathbf{S}\hat{\boldsymbol{\sigma}}_v\| \, \mathrm{d}A = \int_A g(\mathbf{X}) \det(\mathbf{S}) \, \|\mathbf{S}^{-T}\hat{\boldsymbol{\sigma}}_u \times \hat{\boldsymbol{\sigma}}_v\| \, \mathrm{d}A. \tag{6.16}$$

6.3.4 Polynomial Bases and their Degree

The choices of bases for the construction of curve, area, and volume rules are not independent due to two reasons: firstly, the curve or area rules are used to evaluate the flux of polynomials $\mathbf{P}_i$ in the construction of area and volume rules, respectively. Secondly, we use surface rules to evaluate polynomials in spatial coordinates arising from the finite element method, and therefore integrals over $g \circ \hat{\boldsymbol{\sigma}}$ with polynomial g need to be approximated sufficiently well. These observations inform our choice of basis $\{\xi^i \eta^j \zeta^k : i + j + k \leq 4\}$ for our volume rules, and the compatible basis $\{\xi^i \eta^j : 0 \leq i, j \leq 4\}$ for our area rules. For surface rules, we use maximal degree of 5 instead of 4 to account for the additional nonlinearity in the area factor.

6.4 Simulating Cut Elements

To illustrate the use of our rules in solving static elasticity problems on domains enclosed by a B-rep, it remains to discuss how we can accurately represent *interpolated* quantities on an *unfitted* mesh. To this end, we enrich elements which are cut by the boundary.

After discussing standard elements, we highlight how a change of basis for enriched elements (1) enables the use of a *standard* FEM implementation for element energy, force, and tangent stiffness evaluations for cut elements, (2) increases the efficiency of these evaluations, and (3) preserves desirable properties of Lagrange shape functions.

Although we apply our technique to elasticity problems within the scope of this paper, our technical contributions are not limited to a particular PDE.

6.4.1 Standard Elements and Elasticity

Because our simulation mesh consists of regular cuboids, hexahedral elements lend themselves. We use standard Lagrange shape functions $N_i : \mathbb{R}^3 \to \mathbb{R}$ to interpolate an element's undeformed nodes $\mathbf{X}_i$, defining a mapping $\mathbf{X}(\boldsymbol{\xi}) = \sum_i N_i(\boldsymbol{\xi})\mathbf{X}_i$ from natural to physical coordinates. Relying on the same interpolation for the deformed configuration, we define the deformation gradient

$$\mathbf{F}(\boldsymbol{\xi}) = \frac{\partial \mathbf{x}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \left(\frac{\partial \mathbf{X}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \right)^{-1} \quad \text{with} \quad \mathbf{x}(\boldsymbol{\xi}) = \sum_i N_i(\boldsymbol{\xi})\mathbf{x}_i. \tag{6.17}$$

The choice of bases for rule construction depends on the order of our shape functions. It is worth pointing out that the mapping from natural to physical coordinates is *linear* for hexahedral elements, meaning that only the coefficients of the polynomial shape functions change. However, note that, unlike for tetrahedral elements, the deformation gradient is *not* constant for linear N_i.

To compute a static equilibrium, the strain energy density $\Psi(\mathbf{F})$ of a linear or hyperelastic material [SB12, SSB13] is integrated over all hexahedral elements of volume V_e

$$E_{\text{int}}(\mathbf{x}) = \sum_e \int_{V_e} \Psi(\mathbf{F}(\boldsymbol{\xi}))\, \mathrm{d}V \tag{6.18}$$

where the integral over element e only depends on the incident nodal degrees of freedom.

Note that we perform integration in physical and *not* in natural coordinates, hence we transform quadrature points $\mathbf{X}_j$ in the undeformed volume to natural coordinates $\boldsymbol{\xi}_j$. To do so, we subtract the "origin" of the element (vertex closest to the origin for a hexahedron with positive coordinates) from the quadrature point, then scale the resulting vector with the inverse of the side lengths of the hexahedron.

6.4.2 Cut Elements

To integrate the strain energy for elements cut into subvolumes, the extended finite element method introduces additional "enriched" degrees of freedom together with specifically constructed shape functions. Our enrichment strategy is mathematically equivalent to Shifted Sign Enrichment (SSE), which was recently used by Koschier et al. [KBT17] to *explicitly* represent strong discontinuities along detailed, piecewise linear cuts. We will first discuss standard SSE, and then present a reformulation that reduces the computational overhead for (multi-)enriched elements and simplifies implementation.

For elements cut into multiple subvolumes V_j, standard SSE adds enriched shape functions to ensure that the function space is *complete* on *every* subvolume V_j

$$\mathbf{x}(\boldsymbol{\xi}) = \sum_i \left(N_i(\boldsymbol{\xi})\mathbf{x}_i + \sum_{j:V_j \neq V_i} \mathbf{1}_{V_j}(\boldsymbol{\xi}) N_i(\boldsymbol{\xi})\mathbf{x}_i^j \right) \tag{6.19}$$

where $\mathbf{x}_i^j$ are the additional degrees of freedom and $\mathbf{1}_{V_j}$ is the characteristic function that evaluates to one if a point with natural coordinates $\boldsymbol{\xi}$ is contained in V_j and zero otherwise.

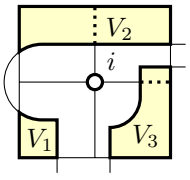
To ensure completeness, the second sum runs over all subvolumes except the one we "assign" the original shape function N_i to, denoted by V_i. Refer to Fig. 6.9 left for a 1D example where two linear basis functions are enriched to ensure that the function space is linear on all three "subvolumes". The formulation used by Koschier et al. is identical to this one except for signs.

Replacing the interpolation of deformed nodes in the deformation gradient with this enriched interpolation, and integrating the strain energy density over the individual subvolumes V_j, we can accurately simulate the elastic response of the body enclosed in a B-rep. However, the traditional approach (Eq. 6.19) of *adding* enriched basis functions to the existing set has practical shortcomings: For enriched elements, the elastic energy depends on the additional degrees of freedom, meaning that the elemental force vector and tangent stiffness matrix grow in size. For example, for linear hexahedral elements cut into k subvolumes, the energy gradient has size $24k$, and the energy Hessian size $24k \times 24k$. Due to the dependence of the elemental energy and energy derivative evaluations on the *number* of subvolumes, custom code is required for every discrete number of subvolumes.

To resolve these practical roadblocks, we propose to perform a change of basis

$$\mathbf{x}(\boldsymbol{\xi}) = \sum_j \left(\sum_i \mathbf{1}_{V_j}(\boldsymbol{\xi}) N_i(\boldsymbol{\xi}) \mathbf{x}_i^j \right). \tag{6.20}$$

As illustrated in Fig. 6.9 right (1D example), this basis spans the same function space, but the support of the basis functions for each subvolume are *local* to their domain. This reformulation has far-reaching benefits. Because we can treat each subvolume V_j of an enriched element like a standard element with nodal degrees of freedom $\mathbf{x}_i^j$ for every N_i, a standard FEM implementation can be used to evaluate the elemental elastic energy and its derivatives. Moreover, unlike the traditional formulation of strong-discontinuity enrichments (Eq. 6.19), the computational complexity of Hessian evaluations scales *linearly* instead of *quadratically* with the number of subvolumes. For example, for linear hexahedral elements, only k evaluations of 24×24 Hessian matrices are needed (instead of one $24k \times 24k$ matrix). Because intersections of CAD models with hexahedral meshes tend to cut elements into large sets of subvolumes, we observe a remarkable increase in simulation and optimization performance. Additionally, while the enriched basis in Eq. 6.19 does *not* fulfill the partition of unity property, our enriched basis does. This guarantees that rigid body movement can be correctly represented even if cut elements are present [Liu16].



It remains to discuss which degrees of freedom $\mathbf{x}_i^j$ are shared between incident elements, and which ones are kept separate: For every element incident to vertex i, we check if the elemental subvolumes are connecting through grid faces. As we illustrate in the inset in 2D, we consider connected subvolumes (volumes separated by dotted lines) to be one entity V_j, adding internal force and tangent stiffness contributions to a *shared* degree of freedom $\mathbf{x}_i^j$. We therefore have three degrees of freedom $\mathbf{x}_i^1$, $\mathbf{x}_i^2$, $\mathbf{x}_i^3$ instead of five in the inset example. For integration, we consider elemental subvolumes separate entities.

6.4.3 Boundary Conditions and Gravity

Unlike conformal meshes, our nodal degrees of freedom do not lie on the surface, and we cannot enforce Dirichlet conditions by holding a subset of them fixed. We instead rely on Nitsche's method [Nit71], a technique well-known in mechanical engineering.

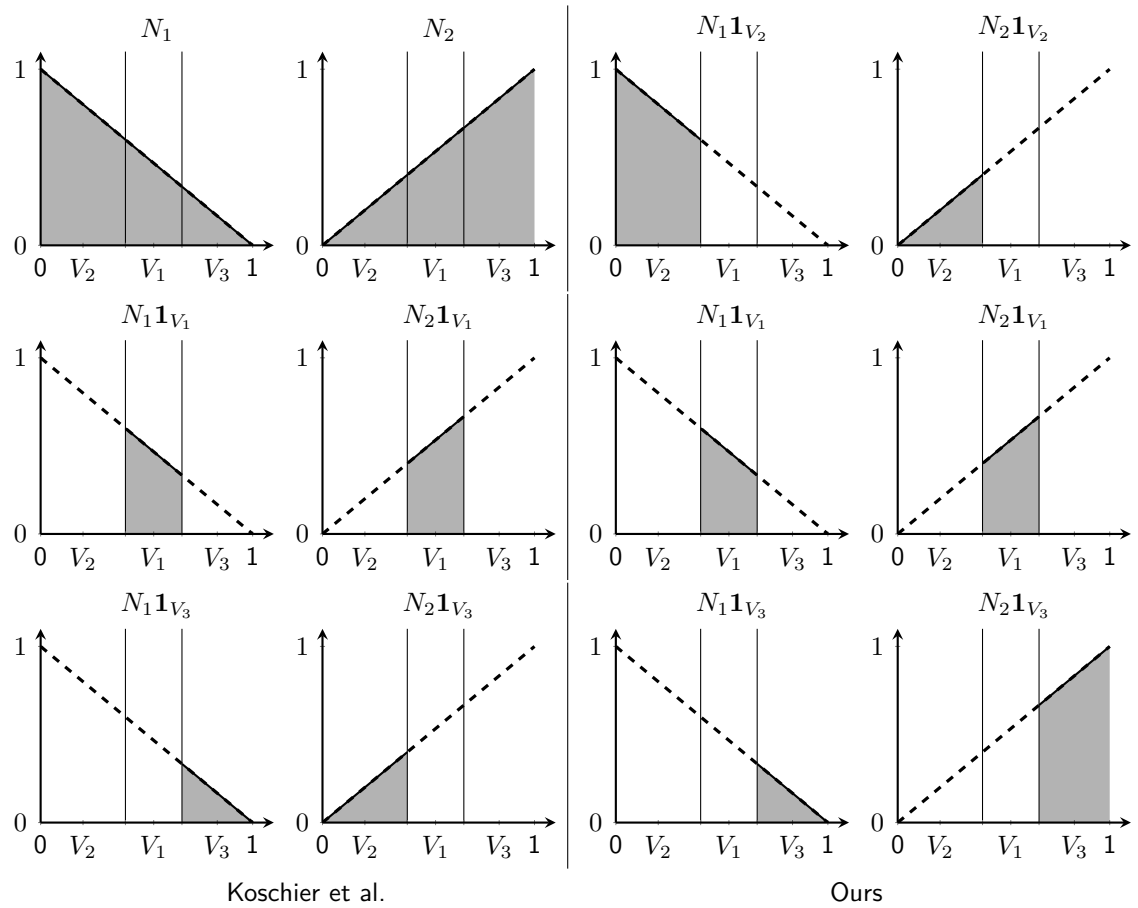


Figure 6.9: **Change of Basis** Comparing our (right) to Koschier et al.'s basis (left) for a 1D example where the domain is partitioned into "subvolumes" V_1, V_2 and V_3, we observe that both are trilinear on all three subvolumes while the support of the individual basis functions is different: while V_2 and V_3 are affected by three, and V_1 by four DOFs for Koschier et al.'s basis, all subvolumes are only affected by two DOFs for our basis.

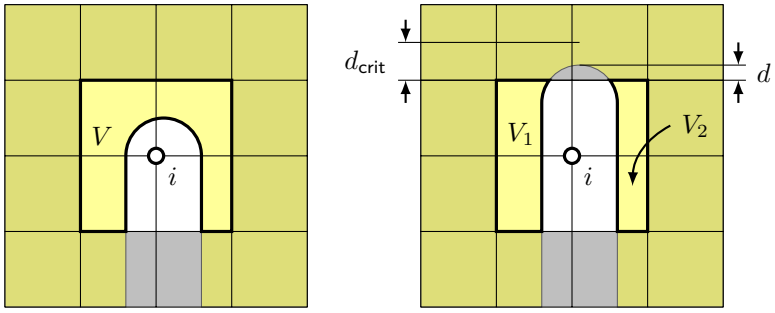


Figure 6.10: **Differentiability** A concave feature of the model (yellow) intersects the hexahedral mesh adjacent to a simulation node i in a single connected component V (left). If the feature moves past the boundary of the adjacent elements, the intersection splits into two parts V_1 and V_2 (right), and the simulation node into two enriched degrees of freedom.

To enforce prescribed displacements $\hat{\mathbf{u}}(\boldsymbol{\xi})$ on a part of the boundary S_{disp}, we introduce a *displacement* energy

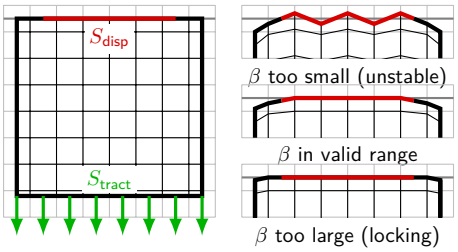
$$E_{\text{disp}}(\mathbf{x}, \boldsymbol{\lambda}) = \int_{S_{\text{disp}}} \boldsymbol{\lambda}(\boldsymbol{\xi}) \cdot (\mathbf{u}(\boldsymbol{\xi}) - \hat{\mathbf{u}}(\boldsymbol{\xi}))\, \mathrm{d}S, \tag{6.21}$$

setting $\mathbf{u}(\boldsymbol{\xi}) = \mathbf{x}(\boldsymbol{\xi}) - \mathbf{X}(\boldsymbol{\xi})$. The Lagrange multiplier *function* $\boldsymbol{\lambda}(\boldsymbol{\xi})$ can be thought of as *reaction traction* acting on the surface S_{disp} to enforce the prescribed displacements. Nitsche showed that $\boldsymbol{\lambda}$ equals $-\mathbf{P} \cdot \mathbf{n}$ where $\mathbf{P} = \frac{\partial \Psi}{\partial \mathbf{F}}$ is the first Piola-Kirchhoff stress (PK1) and $\mathbf{n}$ the surface normal evaluated at $\boldsymbol{\xi}$. This substitution removes $\boldsymbol{\lambda}$ as an unknown variable.

In a discrete setting, this formulation is unstable, and we add the common stabilizer

$$\frac{\beta}{2} \int_{S_{\text{disp}}} \|\mathbf{u}(\boldsymbol{\xi}) - \hat{\mathbf{u}}(\boldsymbol{\xi})\|^2 \, \mathrm{d}S \tag{6.22}$$

with stabilization parameter β, to E_{disp}. If β is chosen too small, the method remains unstable. If it is too large, elements intersecting the boundary S_{disp} effectively lock. Chosen in the right range, β balances the enforcement of prescribed displacements with the elastic response of the model as illustrated in the inset.



Surface tractions $\hat{\mathbf{t}}(\boldsymbol{\xi}) : \mathbb{R}^3 \to \mathbb{R}^3$ that act on another part S_{trac} of the boundary (see above inset) can be added as in standard FEM with a *traction* energy

$$E_{\text{trac}}(\mathbf{x}) = \int_{S_{\text{trac}}} \hat{\mathbf{t}}(\boldsymbol{\xi}) \cdot \mathbf{u}(\boldsymbol{\xi})\, \mathrm{d}S. \tag{6.23}$$

Note that for integration over the domains S_{disp} and S_{trac}, our rules for curved surface domains can readily be used.

Minimizing the total potential energy E with $E_{\text{ext}} = E_{\text{trac}} + E_{\text{disp}}$, we can solve for the equilibrium state $\mathbf{x}$, while accounting for features at subelement resolution. For models where gravity is non-negligible, we add the energy $E_{\text{gravity}}(\mathbf{x}) = \int_V \rho\, \mathbf{g} \cdot \mathbf{u}(\boldsymbol{\xi})\, \mathrm{d}V$ with density ρ to the external energy. The constant 3D vector $\mathbf{g}$ points in the direction of gravity and has magnitude equal to the gravitational acceleration.

6.4.4 Differentiability

In its current form, our XFEM formulation is sufficiently smooth and differentiable with one exception as we illustrate in Fig. 6.10 with an example in 2D: if shape parameters change, a subvolume V could split into two subvolumes V_1 and V_2 in the 1-neighborhood of a vertex i. As a consequence, the degree of freedom $\mathbf{x}_i$ is split into two enriched degrees of freedom $\mathbf{x}_i^1$ and $\mathbf{x}_i^2$, and the elastic response of the model may become discontinuous as a function of the shape parameters. Note that our simulation is only discontinuous at the point of the split.

To detect cases that could lead to a discontinuity, we compare the subvolumes in the 1- and 2-neighborhood of a vertex. If two (or more) subvolumes in the 1-neighborhood are connected in the 2-neighborhood, adjustments to shape parameters could lead to a discontinuity.

To avoid these discontinuities, we define a metric d that is zero at the point where a volume splits, and varies in the range $[0, d_{\text{crit}}]$ if the two subvolumes are close to merging. A metric that lends itself is the shortest orthogonal distance between the boundary of the 1-neighborhood of vertex i and a geometric saddle point in the model (compare with Fig. 6.10 right). To ensure that degrees of freedom smoothly merge as d goes to zero, we add penalties of the form

$$b(d)\|\mathbf{x}_i^1 - \mathbf{x}_i^2\|^2 \quad \text{with} \quad b(d) = \log^2\left(\min\left\{1, \frac{d}{d_{\text{crit}}}\right\}\right) \tag{6.24}$$

to our potential energy E. This procedure guarantees a smooth transition of the simulation result as concave features pass through the simulation mesh. For numerical reasons, we cut off b at a high value, so the conditioning of the discretized PDE does not deteriorate. Note that we have not observed any negative effects due to this choice.

6.5 Optimizing Shape Parameters

Our shape optimization enables a wide range of applications, including combined mass distribution and strength-to-weight ratio, rest shape optimization, and various other inverse design problems that require an accurate integration of properties or discretized PDE equations over the parameterized design domain enclosed by a B-rep.

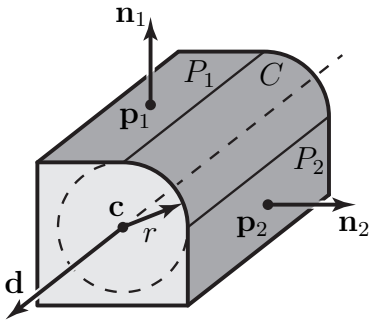
A key advantage of our approach is that our hexahedral simulation mesh is *independent* of our parameterized boundary representation. Hence, in the context of strength-to-weight ratio or rest shape optimization, we can work with the *same* hexahedral mesh, even under large changes of shape parameters.

Before we discuss how to efficiently compute derivatives of integration rules, the elastic response, and the user-specified parameterization, we will refine our parameterization formulation.

6.5.1 Parameterization

A good parameterization should provide sufficient degrees of freedom to enable meaningful improvements of combinations of objectives, while preserving the original design intent of the CAD model. To this end, shape parameters should describe its shape in an intuitive manner, enabling the user to easily select a subset of them for optimization.

On a low level, we represent CAD models as a set of NURBS patches as described in Section 6.2. We collect all NURBS control points in a vector $\mathbf{q}$. CAD models typically contain many geometric primitives, such as, e.g., planar, cylindrical, or toroidal segments. These are more



Constraints(P_1, C):

$$(\mathbf{p}_1 - \mathbf{c}) \cdot \mathbf{n}_1 - r = 0$$
$$\mathbf{d} \cdot \mathbf{n}_1 = 0$$

Constraints(P_2, C):

$$(\mathbf{p}_2 - \mathbf{c}) \cdot \mathbf{n}_2 - r = 0$$
$$\mathbf{d} \cdot \mathbf{n}_2 = 0$$

Figure 6.11: **Tangential Constraints** A cylindrical segment C is tangent to two adjacent planes P_1 and P_2. To maintain the tangencies, constraint equations are generated and added to $\mathbf{c}_{\text{para}}$. The surface parameters $\mathbf{p}_1, \mathbf{n}_1, \mathbf{p}_2, \mathbf{n}_1, \mathbf{c}, r, \mathbf{d}$ are part of the parameter vector $\mathbf{p}$.

intuitively described using mid-level parameters such as, e.g., the central axis and radius of a cylinder. We collect all these parameters in a vector $\mathbf{p}$. Control points in $\mathbf{q}$ necessarily depend on the values of parameters in $\mathbf{p}$, e.g., changing the radius of a cylinder moves the NURBS control points describing the cylinder's geometry. We denote this mapping as $\mathbf{q} = \mathbf{q}(\mathbf{p})$. By only modifying $\mathbf{q}$ indirectly through $\mathbf{p}$, primitives retain their basic shape.

However, additional constraints are required to preserve the design intent of a CAD model. A frequent example is a fillet, i.e., an initially sharp edge that has been rounded off by adding a cylinder segment, as seen in Fig. 6.11. The cylinder C is tangent to the planes P_1 and P_2, and this tangency must be preserved when modifying $\mathbf{p}$; otherwise, unintended sharp edges may be introduced. Constraints of this type can be formalized as equations in $\mathbf{p}$, as exemplified for the fillet in Fig. 6.11.

Once a CAD model is loaded, we traverse all pairs of adjacent surfaces and detect tangency relationships that need to be preserved. This is done automatically using a look-up table that stores possible relationships between primitive types, such as the one between a cylinder and a planes in the fillet example above. Whenever a situation like this is encountered, a list of implicit constraints is generated for the pair of surfaces in question. Processing the entire model in this manner yields a non-linear constraint system $\mathbf{c}_{\text{para}}(\mathbf{p}, \mathbf{q}(\mathbf{p})) = \mathbf{0}$ that needs to remain satisfied throughout optimization. In addition to these automatic constraints, our UI enables the user to define high-level model parameters by grouping specific parameters in $\mathbf{p}$, to keep certain surfaces fixed, or to enforce symmetries in the model. These additional constraints are also added to $\mathbf{c}_{\text{para}}$.

During optimization, it may occur that no value of $\mathbf{p}$ exists which satisfies the tangency and user-specified constraints in $\mathbf{c}_{\text{para}}$ at the same time. To ensure that the CAD model remains valid regardless, we introduce *effective* shape parameters $\mathbf{p}_\perp$, which represent the projection of the given parameters $\mathbf{p}$ onto the constraint manifold defined by $\mathbf{c}_{\text{para}}$. This projection together with $\mathbf{q} = \mathbf{q}(\mathbf{p}_\perp)$ defines an *implicit* mapping from model parameters to NURBS control points. To summarize, we minimize

$$f_{\text{para}}(\mathbf{p}_\perp, \mathbf{q}) = \frac{1}{2}\|\mathbf{p}_\perp - \mathbf{p}\|^2 + \frac{1}{2}\|\mathbf{q}(\mathbf{p}_\perp) - \hat{\mathbf{q}}\|^2 \tag{6.25}$$

over the constraint manifold spanned by $\mathbf{c}_{\text{para}}$, or the corresponding Lagrangian

$$\mathcal{L}(\mathbf{p}_\perp, \mathbf{q}, \boldsymbol{\lambda}) = f_{\text{para}}(\mathbf{p}_\perp, \mathbf{q}) - \boldsymbol{\lambda}^T \mathbf{c}_{\text{para}}(\mathbf{p}_\perp, \mathbf{q}(\mathbf{p}_\perp)) \tag{6.26}$$

to first-order optimality. Here, $\mathbf{p}$ is the set of shape parameters modified through an optimization step, and $\mathbf{p}_\perp$ is its projection onto $\mathbf{c}_{\text{para}}$. Insufficiently constrained control points $\mathbf{q}(\mathbf{p}_\perp)$ are modified as little as possible by keeping them close to their initial values $\hat{\mathbf{q}}$.

6.5.2 Shape Optimization and Derivatives

After the user specifies a desired parameterization together with a set of objectives that depend on the parameterization and the elastic response of the model, we aim at solving the first-optimality constrained problem

$$\min_{\mathbf{p}} f(\mathbf{p}_\perp(\mathbf{p}), \mathbf{q}(\mathbf{p}_\perp), \mathbf{x}(\mathbf{p}_\perp)) \text{ s.t. } \begin{bmatrix} \mathcal{L}_{\mathbf{p}_\perp} \\ \mathcal{L}_{\mathbf{q}} \\ \mathcal{L}_{\boldsymbol{\lambda}} \end{bmatrix} = 0 \text{ and } E_{\mathbf{x}} = 0. \tag{6.27}$$

Note that, compared to our formulation as outlined in our overview (Section 6.2), we replace the direct dependence of the objective on the parameters with an implicit dependence $\mathbf{p}_\perp(\mathbf{p})$. Posing a design optimization in this particular way is advantageous because, during numerical optimization, the parameters $\mathbf{p}$ can take on values that do not fulfill our parameterization constraints, for example during line search along a descent direction. The combination of a continuous "projection" $\mathbf{p}_\perp(\mathbf{p})$, formulated with a first-order optimality constraint on a parameterization Lagrangian, and the use of only valid sets of parameters in objective evaluations, enables the use of a standard quasi-Newton for optimization where first-order optimality constraints are *implicitly* enforced.

In objective and objective gradient evaluations for a particular $\mathbf{p}$, we first solve the Lagrangian to first-order optimality. We then use the resulting set of control points $\mathbf{q}(\mathbf{p}_\perp)$ in the minimization of the potential energy E to equilibrium, $E_{\mathbf{x}}(\mathbf{q}(\mathbf{p}), \mathbf{x}(\mathbf{p})) = 0$. To compute shape derivatives of our objective with respect to shape parameters

$$\mathsf{d}_{\mathbf{p}} f = f_{\mathbf{p}_\perp} \mathsf{d}_{\mathbf{p}} \mathbf{p}_\perp + f_{\mathbf{q}} \mathsf{d}_{\mathbf{p}} \mathbf{q} + f_{\mathbf{x}} \mathsf{d}_{\mathbf{p}} \mathbf{x}, \tag{6.28}$$

we apply the implicit function theorem to our parameterization

$$\begin{bmatrix} \mathcal{L}_{\mathbf{p}_\perp,\mathbf{p}_\perp} & \mathcal{L}_{\mathbf{p}_\perp,\mathbf{q}} & \mathcal{L}_{\mathbf{p}_\perp,\boldsymbol{\lambda}} \\ \mathcal{L}_{\mathbf{q},\mathbf{p}_\perp} & \mathcal{L}_{\mathbf{q},\mathbf{q}} & \mathcal{L}_{\mathbf{q},\boldsymbol{\lambda}} \\ \mathcal{L}_{\boldsymbol{\lambda},\mathbf{p}_\perp} & \mathcal{L}_{\boldsymbol{\lambda},\mathbf{q}} & \end{bmatrix} \begin{bmatrix} \mathsf{d}_{\mathbf{p}} \mathbf{p}_\perp \\ \mathsf{d}_{\mathbf{p}} \mathbf{q} \\ \mathsf{d}_{\mathbf{p}} \boldsymbol{\lambda} \end{bmatrix} = - \begin{bmatrix} \mathcal{L}_{\mathbf{p}_\perp,\mathbf{p}} \\ \mathcal{L}_{\mathbf{q},\mathbf{p}} \\ \mathcal{L}_{\boldsymbol{\lambda},\mathbf{p}} \end{bmatrix} \tag{6.29}$$

and quasi-static equilibrium $E_{\mathbf{x},\mathbf{x}} \mathsf{d}_{\mathbf{p}} \mathbf{x} = -E_{\mathbf{x},\mathbf{q}} \mathsf{d}_{\mathbf{p}} \mathbf{q}$ where we use $(.)_{\mathbf{p}}$ for partial, and $\mathsf{d}_{\mathbf{p}}(.)$ for total derivatives. For efficiency, we rely on the adjoint method.

6.5.3 Taking Derivatives of Quadrature Rules

If we make adjustments to our shape parameters, the volume V changes, and hence the domains of our hierarchical rules. While these changes are restricted to elements that were or are newly cut by the boundary, the cost of taking derivatives of rules can be considerable and requires a significant amount of bookkeeping. To reduce the computational complexity and simplify the implementation of shape derivatives, we describe how we can avoid some of the terms in our volume, area, and surface rules. Note that rule construction only depends on the control points of the B-rep, hence we can safely ignore other dependencies here.

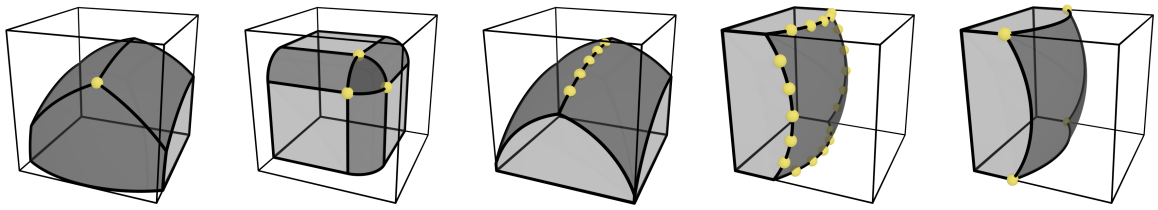


Figure 6.12: **Derivatives of Curve Rules** To parameterize curve integrals, we sample intersection curves between patches and hexahedral grid planes. We differentiate the following cases (from left to right): intersection points between three or more model surfaces (cases 1 and 2), sample points on an intersection curve between two surfaces (case 3), sample points on surface-grid intersection curves (case 4), and intersection vertices between a surface and two grid planes (case 5).

Area, Surface, and Volume Rules To keep the matrix $\mathbf{A}$ in the moment fitting equation (Eq. 6.11) constant, we propose to transform the non-standard domains. However, the right-hand side $\mathbf{b}$ depends on the shape of the non-standard domain, hence the quadrature points and weights, in general, depend on the shape parameters

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}} \int_{D(\mathbf{q})} g(\mathbf{X}) \, \mathrm{d}D = \sum_j \left(\frac{\mathrm{d}w_j}{\mathrm{d}\mathbf{p}} g(\mathbf{X}_j) + w_j \frac{\partial g(\mathbf{X})}{\partial \mathbf{X}} \frac{\mathrm{d}\mathbf{X}_j}{\mathrm{d}\mathbf{p}} \right). \tag{6.30}$$

A key observation to increase the efficiency of rule derivatives, is, that, if we keep the transformations for volume, area, and surface integrals *after initial* rule construction fixed, the quadrature points *no longer* depend on the shape parameters for volume, area, and surface rules

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}} \int_{D(\mathbf{q})} g(\mathbf{X}) \, \mathrm{d}D = \sum_j \frac{\partial w_j}{\partial \mathbf{p}} g(\mathbf{X}_j) \quad \text{for} \quad D \in \{A, S, V\}. \tag{6.31}$$

Note that for edge and curve rules, the transformation from the general domain $[a, b]$ to a standard domain $[0, 1]$ is necessary. Otherwise, we cannot apply tabulated Gauss-Legendre rules. However, if we keep applying the initial transformation for area, surface, and volume rules, the weights, computed with the moment fitting equation, only depend on the right-hand side $\mathbf{b}$ but not on a shape-dependent transformation. Crucially, these shape derivatives are *exact* if function g is in the function space spanned by the bases used for rule construction.

Curve Rules As we pointed out in Section 6.3, it is, in general, not possible to extract an analytical parameterization for intersection curves that arise when several NURBS patches intersect within a hexahedral element, or a NURBS intersects with one of the element planes. Hence, we represent these planar or spatial curves with sample points, and differentiate between several cases, illustrated in Fig. 6.12.

During optimizations, we make changes to shape parameters, and implicitly also to control points. Changes to the control points, in turn, move the position of sample points on intersection curves. To treat NURBS patches and element planes the same, we parameterize the latter, defining a mapping from parameter values $\mathbf{u} = [u, v]^T$ to plane points $\hat{\boldsymbol{\sigma}}(\mathbf{u}) \in \mathbb{R}^3$. A sample point on two or more "surfaces" is then defined by a pair of uv-coordinates for each surface. To be able to take shape derivatives, it is important to understand the relationship between these coordinates and the shape parameters.

To this end, it is best to look at a specific example (case 1 or 5 in Fig. 6.12) where three "surfaces" intersect in a single point: if we change $\mathbf{p}$, the control points of the three patches and also the uv-coordinates in their respective parameter domain, change. What uniquely defines the uv-coordinates is the constraint that they all map to the *same* point in 3D, formalized with an equation $\hat{\boldsymbol{\sigma}}_i(\mathbf{u}_i(\mathbf{p}), \mathbf{q}(\mathbf{p})) - \hat{\boldsymbol{\sigma}}_j(\mathbf{u}_j(\mathbf{p}), \mathbf{q}(\mathbf{p}))$ for every pair (i, j) of "surfaces". Collecting these equations in a system $\boldsymbol{\Sigma} = 0$, and the uv-coordinates in a vector $\mathbf{U}$, we apply the implicit function theorem to compute analytical derivatives $\mathrm{d}_\mathbf{p}\mathbf{U} = -\boldsymbol{\Sigma}_\mathbf{U}^{-1}\boldsymbol{\Sigma}_\mathbf{q}\,\mathrm{d}_\mathbf{p}\mathbf{q}$.

However, this particular case is rare because real-world CAD models typically have filleted edges and corners with either all adjacent surfaces or subsets of them being tangent (case 2). In such cases, the Jacobian $\boldsymbol{\Sigma}_\mathbf{U}$ becomes rank-deficient. Another case that leads to a rank-deficiency in $\boldsymbol{\Sigma}_\mathbf{U}$ arises if we place sample points on intersection curves that are defined by *two* "surfaces" (case 3 or 4 in Fig. 6.12).

To be able to compute derivatives in these cases, we first classify sample points by analyzing the normals of adjacent surfaces. We then complement the equations in $\boldsymbol{\Sigma} = 0$ with planes that span the null space. For example, for a sample point on a sharp edge (cases 3 and 4), we define a plane whose normal is set to the cross product of the two surface normals. Because the components of derivatives that lie in this null space do not change the value of our integrals to first order, we can safely ignore them after computing the derivatives.

6.6 Results

Accuracy of Quadrature We evaluated the accuracy of our quadrature scheme by comparing the results to a ground truth and to the results obtained with Müller et al.'s method [MKO13]. For evaluation, we intersected the model shown in Fig. 6.4 with a regular grid to yield a large number of geometric subvolumes, and selected a set of 23 test monomials. The ground truth was obtained by finely triangulating every subvolume and analytically integrating every monomial.

We generated one integration rule for each subvolume using our method and one using Müller et al.'s method. For a fair comparison, we used the same quadrature point locations on surfaces and in volumes, and the same polynomial basis. As an error metric, we use the absolute difference between the ground truth integral and the quadrature solution, divided by the integral over the surrounding element. All combinations of subvolumes and monomials yield $18,078$ data points in total.

Fig. 6.4 shows that our method yields a worst-case relative error of 8.6×10^{-4}, with the vast majority of samples below 10^{-4}. Müller et al.'s method may fail to produce usable rules on curved surfaces if the volume contains few planar surfaces (compare with Fig. 6.6), or if the distribution of quadrature points on curved surfaces is unfavorable.

Convergence of Quadrature Fig. 6.13 shows additional tests for the convergence of integration results. We performed volume integral tests on an analytical surface (a semi-torus, m1), and a model with small topological features (m2). Three parameters were varied during the test: the tessellation level h of the regular grid, the number of segments per curve used to approximate line integrals (cf. Section 6.3.1), and the order of these segments (linear or parabolic).

At the lowest tessellation level, the models are embedded in grids of $6 \times 12 \times 4$ cells for m1 and $10 \times 1 \times 6$ cells for m2, respectively. For m2, this is sufficiently coarse that individual cells

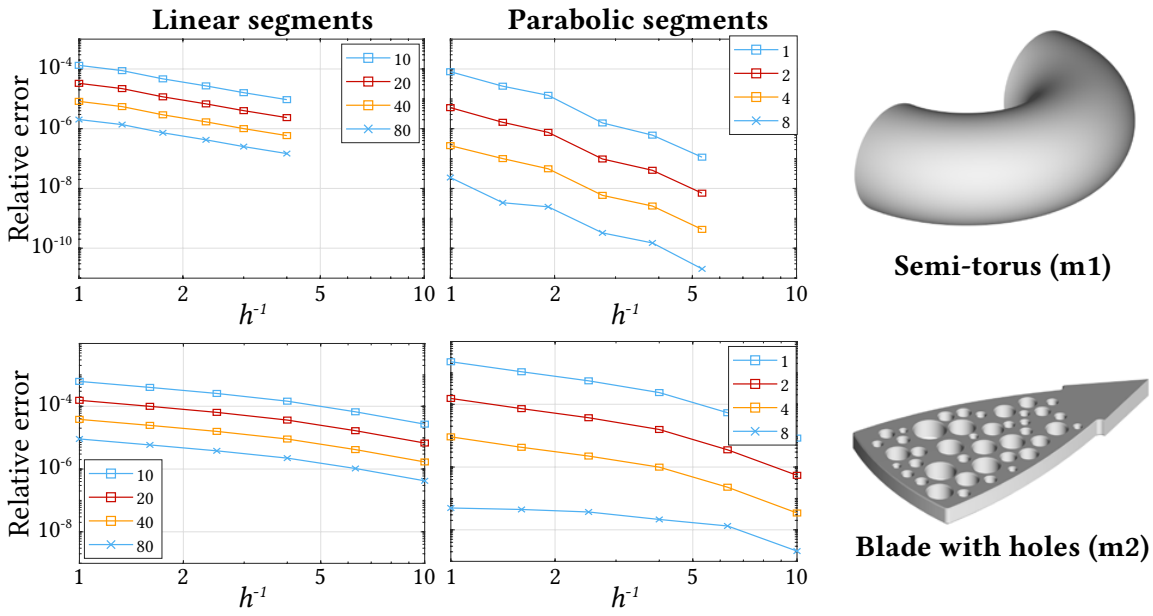


Figure 6.13: **Quadrature convergence.** Relative error of volume integrals under refinement of integration grid and refinement of curve integral approximation. The cell side length in the integration grid is given by h, and is isotropically refined in all three directions. Each curve (cf. Section 6.3.1) is split into a constant number of segments, which is given in the legend. The graphs compare a piecewise-linear curve approximation and a piecewise-parabolic curve approximation. Convergence is faster for parabolic segments, and accuracy improves with both types of refinement.

may contain several holes. Relative integration errors of down to 10^{-7} may still be reached by increasing the number of segments per curve. The comparison between segment types shows that parabolic segments are more economical; two parabolic segments achieve approximately the same accuracy as twenty linear segments. The diagrams also indicate faster convergence rates for parabolic segments, which matches theoretical predictions [AV93]. The computational cost per segment is similar for linear and parabolic segments, as they use the same number of integration points.

Comparison to Standard FEM We performed a simulation test of our XFEM formulation on a standard example of a ring in compression. This was to verify that the extended formulation with Nitsche's method on Dirichlet conditions converges to the same result as standard FEM. A front view of the ring model is shown in Fig. 6.14 (top, right). The blue bars denote Dirichlet conditions, with an enforced compressive displacement on the bottom. Vertical stress, σ_y, along a horizontal cross section is shown as a red curve.

To ensure stability under different embeddings, the simulation was repeated for eight different rotations of the model within the grid. This leads to Dirichlet conditions and stress measurements at different inclinations, as indicated in Fig. 6.14 (bottom, right). The four plots on the left show XFEM stresses for different *h*-refinement levels, where $h = 1$ corresponds to the tessellation shown in the figure. The ground truth obtained from high-resolution standard FEM is shown as the red curve, and the measurements from differently rotated XFEM meshes are shown as gray curves. The plots show that the stresses converge consistently for all rotations.

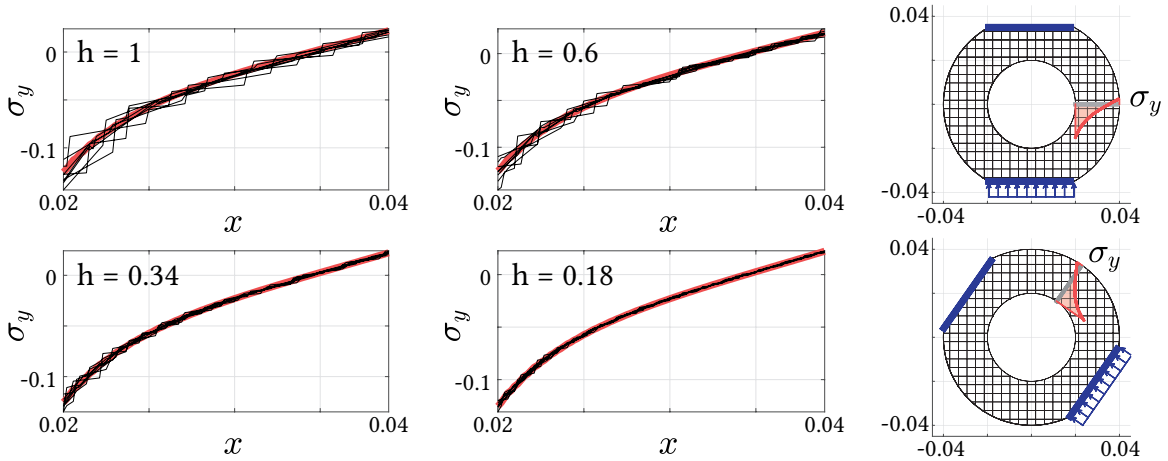


Figure 6.14: **Stress convergence.** Simulation of ring in compression using XFEM, embedded in the simulation grid in different orientations (right). Vertical stresses along horizontal cross-section (left) for different h-refinement levels and different rotations (gray curves), and ground truth (red curve).

Aircraft Our differentiable simulator can deal with models of high geometric complexity, exemplified by a model of the internal structure of an aircraft where we simulate the deflection of the wing under the load of the aircraft (Fig. 6.1 left).

6.6.1 Shape Optimization

In the remainder of this section, we will discuss our optimization results. To visualize parameterizations, we use yellow for parameterized surfaces and orange for surfaces that move due to tangent preservation constraints. Dirichlet conditions are shown in red and surface tractions in green.

Wrench The wrench model (see Figs. 6.2 and 6.17 top row) is composed of a set of NURBS patches that are subject to many tangency relationships along its edges. The model is parameterized by the height and width of the slotted indentation in the handle, the height and width of the hole in the handle, and the fillet radius around the hole. Assigning a linear elastic material and parameters for a standard tool steel, we optimize the strength-to-weight ratio under a mechanical advantage setting where a torque is applied to a nut.

Our optimization co-optimizes a type one objective (Eq. 6.4), whose integrand is set to the exponentiated distance of the Cauchy stress to the von Mises failure surface [SZB18], and a type two objective (Eq. 6.5) integrating the model's density. As we can see in Fig. 6.17, our shape optimization succeeds in significantly improving the mode's strength, with the handle hole geometry smoothly passing through several hexahedral elements. Note that our parameterization preserves all tangencies.

Asymmetric Wheel and Fidget Spinner While many different designs for automobile wheels exist, they are typically rotationally symmetric in order to ensure that they will spin stably around their axis and not introduce vibrations. A combination of a type one (Eq. 6.4 with integrand set to Ψ for linear elasticity) and a set of type two objectives (Eq. 6.5; integrals over all monomials) enables the co-optimization of the strength-to-weight ratio and mass distribution of an *asymmetric* wheel design (see Fig. 6.1 middle and Fig. 6.15). As we demonstrate in

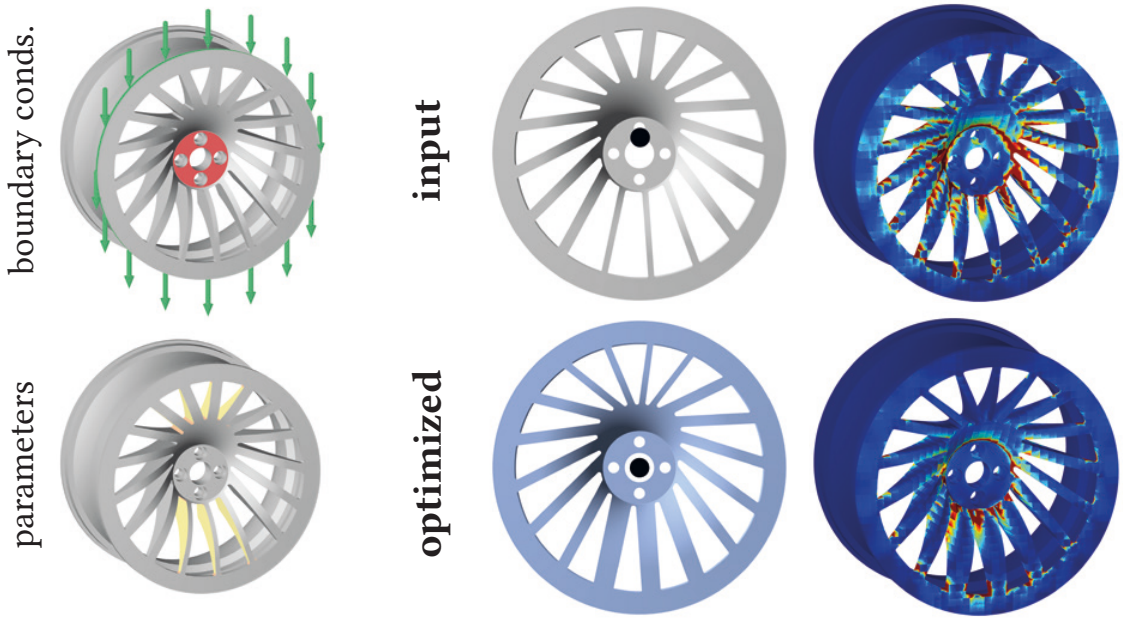


Figure 6.15: **Asymmetric Wheel.**

the accompanying video with scaled-down, 3D printed wheels, the unoptimized input wobbles while our optimized design spins stably. The reason for this difference is that the center of mass (black spheres in the second column in Fig. 6.15) does not lie on the spinning axis (9.9 mm off for a wheel of 40 cm diameter; 22 mm off when considering the mass properties of the wheel spokes and hub only) for the unoptimized design. The parameterization used in the optimization allows for the width of the spokes of the wheel to vary, and the additional compliance term ensures that the result is not only functional (CoM misalignment $<$0.1 mm) but also optimal from a structural point of view (see stress visualizations in Fig. 6.15). While structural considerations are less relevant in toy design, the use of only type two objectives enables the design of asymmetric spinning toys such as a fidget spinner that combines (with some imagination) a teapot, a SIGGRAPH logo, and a bunny in a unified design (see Fig. 6.17 bottom row).

Motor Housing In this example, we optimize the compliance of an aluminium housing for an electric motor, with cooling fins on the outside. Tailored for manufacturing by casting or machining, it is of paramount importance to keep the model undercut-free. Moreover, to preserve its function and appearance, it is essential to enforce rotationally symmetric changes. To this end, we parameterize the thickness and width of the 6 "spokes" with a total of 4 parameter (Fig. 6.16 params), constraining the normals of the surfaces that are orthogonal to the symmetry axis in the initial design, to remain orthogonal throughout optimizations. Relying on a type one objective with Ψ set to linear elasticity, we are able to reduce the overall compliance of the model by 6.7%.

Lampshade Targeting furniture design, we perform hyperelastic rest shape optimization [CZXZ14] on an initially flat lampshade design that consists of six rotationally symmetric, curved blades protruding from the central unit containing the light fitting (see Fig. 6.1 right and Fig. 6.17 middle row). In contrast to our other examples, we parameterize the *continuous* outer surface of the lampshade design (a total of 48 control points) and optimize the model's rest shape (type three objective, Eq. 6.6) such that the outer surface deforms into a toroidal

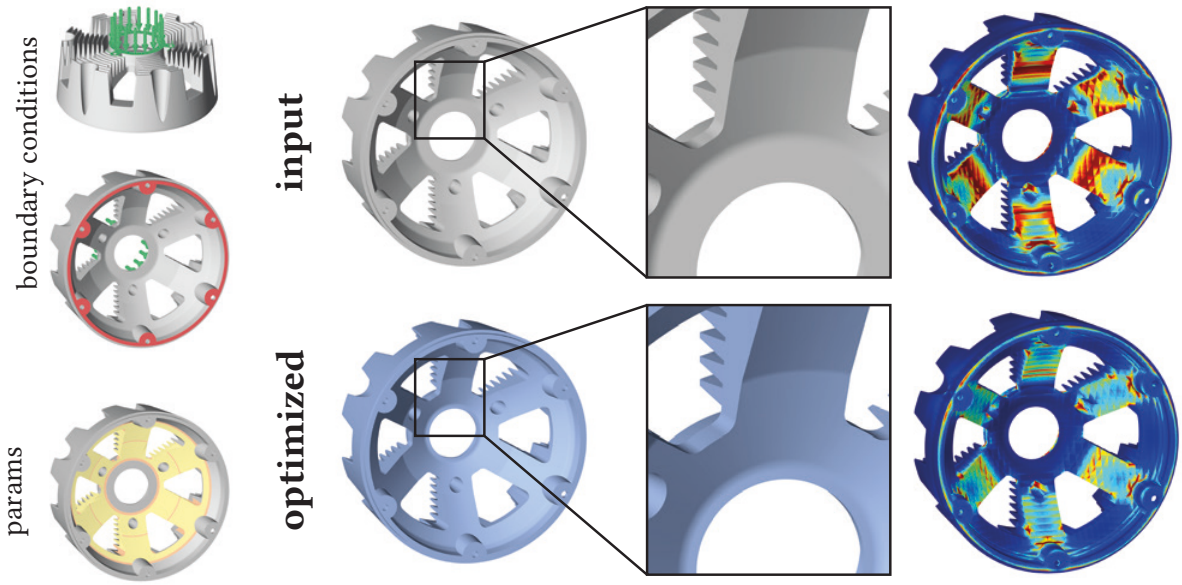


Figure 6.16: **Motor Housing.**

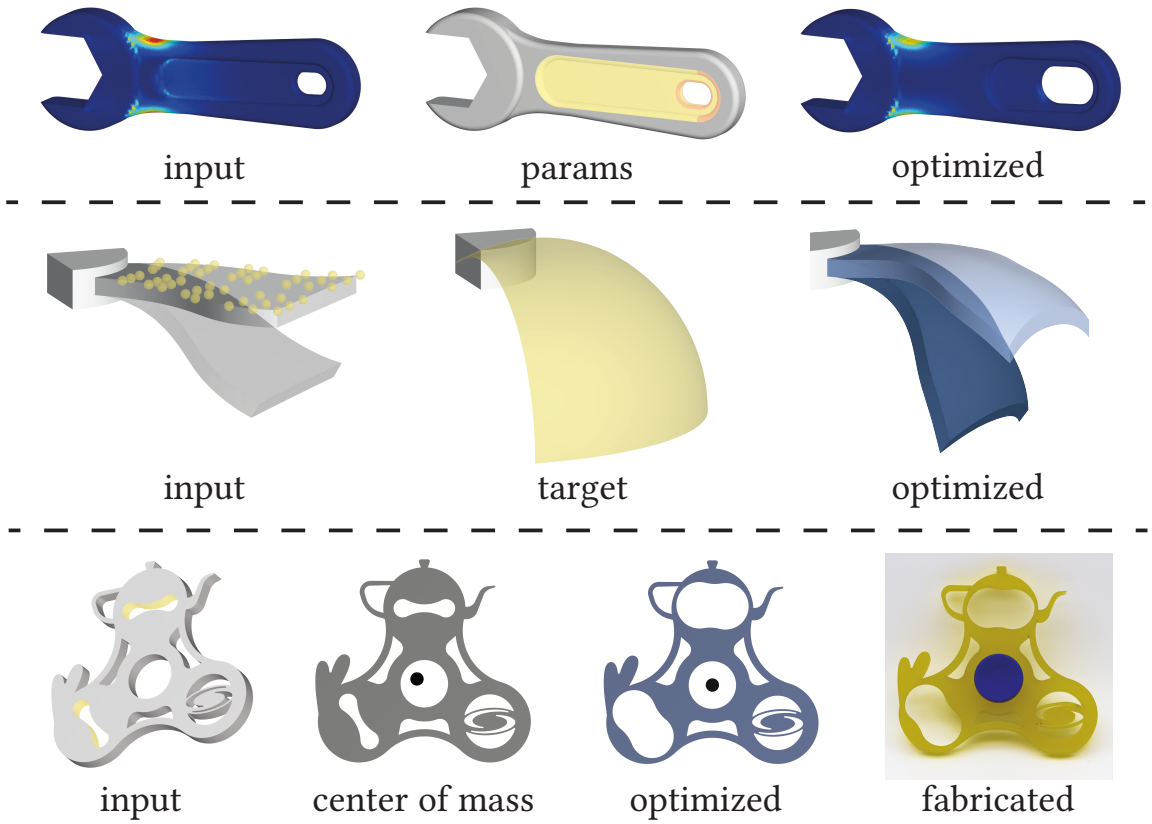


Figure 6.17: **Wrench, Lampshade, and Fidget Spinner.**

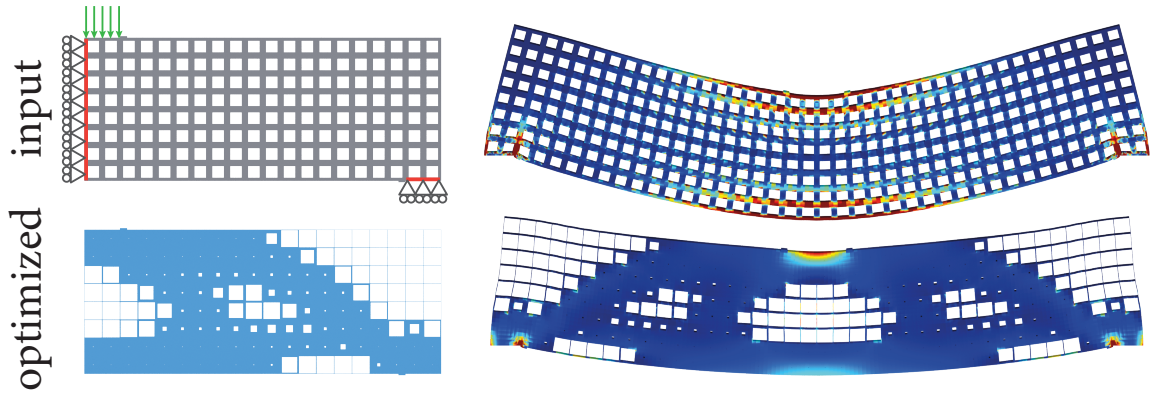


Figure 6.18: **Topology Optimization.**

target shape under self-weight. During optimizations, we keep the curvy silhouette fixed, and constrain the inner surface to move with the outer surface with a thickness preservation constraint. To manufacture the optimized lampshade, we use MoldStar 30 rubber. As can be seen in our accompanying video and in Fig. 6.17, the lampshade deforms into the desired target shape (max. target matching error is 16 mm before and 0.6 mm after optimization). Because Schulz et al.'s technique [SXZ+17] does not scale well with the number of parameters, shape optimizations like this one would be difficult to perform with their method. For accurate simulations, we rely on a Mooney-Rivlin model.

Topology Optimization Although our framework does not currently support topology changes, we can mimic topology optimization by optimizing a grid of 20×8 square holes (total of $n = 160$ shape parameters). The objective is to design a symmetric bridge that minimizes compliance under a volume constraint. The initial grid with boundary conditions, the optimized grid, and the deformed states are shown in Fig. 6.18.

The side length l_i of any square hole is allowed to vary within $0 < l_i < 1$. The true volume fraction of a geometric cell is given by $V_i = 1 - l_i^2$. To drive parameters towards extremal values, we use $\tilde{V}_i = V_i^{2/3}$ to penalize the volume of intermediate cells relative to their stiffness, and constrain $\frac{1}{n}\sum_i \tilde{V}_i = 0.7$. The simulation uses an 80×32 grid, so each cell is meshed by a 4×4 block of elements, some of which appear and disappear as walls move through them. The

Demonstration	Grid	#SV	#P	#S	t	#i
Aircraft	160x40x160	15,840	-	679	104	-
Wrench	32x4x12	926	5	54	8.0	64
Fidget Spinner	16x16x1	161	12	82	3.2	28
Asym. Wheel	10x20x20	1,652	14	154	70	31
Motor Housing	30x30x18	6,628	4	557	86	43
Lampshade	26x26x38	1,954	48	6	352	17
Topology Opt.	80x1x32	2,143	160	653	50	33

Table 6.1: **Simulation and Optimization Statistics** Columns from left to right: name of the demonstration; resolution of simulation grid; number of integration subvolumes (#SV); number of shape parameters (#P); number of NURBS patches in the CAD model (#S); time per iteration or total simulation time (Aircraft) in seconds (t); number of iterations (i).

result is in line with conventional topology optimization results. Moreover, this demonstrates the capability of our method to deal with large numbers of optimization parameters as well as very thin model sections.

Performance All simulations and optimizations were performed on a single core of an Intel Core i7-8700. Tab. 6.1 provides data about the complexity of models and timings of optimization and simulation routines. We rely on *linear* shape functions in all our simulations.

6.7 Conclusion

We have devised a generic shape optimization that enables the solution of a wide range of computational design problems directly on a CAD representation. While we have developed our hierarchical integration and extended finite element formulation with an eye on shape optimization, we see applications beyond the discussed context. For example, because our simulation of CAD models is fully *differentiable*, it is well suited for applications in geometric deep learning [KMJ+19].

Our proposed change of basis of enriched shape functions makes it straightforward to turn an existing FEM implementation into one that supports strong discontinuities in cut elements. Moreover, it makes Hessian computations for cut elements significantly more efficient, and leads to a basis that preserves not only the Kronecker delta but also the partition of unity property. A change of basis as used herein would be beneficial in XFEM applications in general.

Furthermore, we introduced a parameterization Lagrangian that enables optimization with a set of parameters that does not lie on the constraint manifold defined by a user-specified set of constraints. *Implicitly* enforcing the first-order optimality of this Lagrangian in shape optimizations, we can compute analytical gradients of the *continuous* projection of the parameters onto the closest set of valid values. We see utility of this projection in other computational design problems and beyond.

Limitations & Future Work While our integration schemes and XFEM formulation supports the use of relatively coarse hexahedral meshes, Dirichlet conditions cannot be fulfilled *exactly* along the boundary of the domain. Although we have not experienced any difficulties in choosing a stabilization parameter that prevents locking or instabilities, striking for the right balance between fulfillment of displacement constraints and elastic response may be challenging. If the fulfillment of constraints is unsatisfactory, an increase in resolution, hence a decrease of computational efficiency, is unavoidable. While we rely on an *axis-aligned* hexahedral grid for increased performance, the support of conforming elements in cells that intersect the boundary of a constrained domain, or an adaptive refinement as common in Finite Cell (FC) methods are valid alternatives left for future work. Another exciting future avenue is the extension of our strong discontinuity formulation to weak discontinuities, enabling applications in dual or multimaterial modeling [VWRKM13]. For example, Nitsche's method could be used to treat discontinuities in the deformation gradient.

Finally, our shape optimization does not handle topological changes in the relationships between neighboring NURBS patches. Enforcing their preservation with constraints, we demonstrate with our results that large changes to shape parameters are possible. While it is often desirable to preserve these relationships, there are applications where topological changes are beneficial, constituting an exciting future direction. A differentiated CAD kernel could be useful in this context [MBA+18].

CHAPTER 7

Conclusion

We explored two topics in fabrication-aware shape optimization and found solutions that operate directly on the same parametric representations that are used in the manual design process. This places these methods firmly in a modern iterative design workflow, in which user editing and simulation-based optimization can alternate and complement each other.

The problem we studied first explores the design of bending-active rod structures in the plane and in three dimensions. We discovered that despite the non-linearity present in computing equilibrium states of these systems, the inverse problems can all be posed as linear programs and solved to optimality almost instantaneously. Furthermore, these problems admit concise geometric descriptions of the design space, some of which are intuitive enough that visual inspection suffices to establish whether a problem instance is feasible.

We also used these geometric characterizations as part of iterative optimization routines that improve the ease of fabrication and stability of designs. For these optimization problems, we arrived at a parametric, spline-based description of curves naturally, because the constraints that define the feasible space depend on the precise location of inflection points, which are difficult to control in the discrete setting.

In the last problem, which studies simulation and shape optimization techniques on CAD models, maintaining and optimizing the original parametric representation was a conscious decision rather than a natural outcome. Most numerical methods for solving PDEs depend on a decomposition of the domain into geometrically simple elements, so it is more straight-forward to discretize CAD models before performing a simulation. We developed a method in which the embedding space is discretized instead of the model, and we perform numerical integration on subdomains bounded by patches of the original parametric surfaces. This way, we achieve a differentiable relationship between CAD parameters and analysis result, which allows us to embed the simulation routine in a gradient-based shape optimization procedure. Thereby we enable a unique workflow in which a designer can manually edit a CAD model, improve it using simulation-based optimization, and go back to making manual adjustments as desired.

The current implementation of this method suffers from a few shortcomings that are inherited from other extended finite element method type simulations. One issue is the application of Dirichlet boundary conditions, which is subject to locking artifacts not present in conforming finite element simulations. While some methods have been proposed to resolve this problem, they usually require model-specific parameter tuning or rely on mixed finite element methods that drastically increase the size of the simulation problem. Another issue is the lack of

adaptive remeshing in our method, which limits its application to coarse simulations, or situations in which the refinement points are fixed a-priori. Preserving differentiability of the simulation result throughout remeshing is likely a formidable challenge, for which a yet unexplored combination of extended finite element techniques and differentiable meshing may be necessary [RAM+21].

The work on elastic curves and Kirchhoff rods presented in this thesis arose from the desire to contribute to fabrication-aware optimization a result that furthers our understanding of the physical limits of a system, rather than explore them heuristically with numerical optimization techniques. For this purpose, I chose to study plane elastic curves as one of the simplest non-linear physical systems that can be reproduced in the real world with relative ease. It became clear that the design space associated with this system is much easier to describe than suspected, and the same turned out to be true for Kirchhoff rods, despite the complexity added by non-linear constraints.

Working on these subjects makes me very optimistic that similar techniques may be fruitful to characterize the design space of physical systems in more than one dimension. One such existing result concerns elliptic PDEs and says that a scalar coefficient field can be reconstructed from the solution and right-hand side under certain sufficient conditions [Ric81]. Recently, Wang et al. [WS21] pointed out a set of necessary conditions for solutions to elliptic PDEs with general positive-definite coefficient matrix fields, but do not discuss to which extent they are sufficient. Studying the latter question may provide valuable insights into design problems, and could possibly be extended to study inverse problems in elasticity.

One such problem, whose solution would have immediate applications to metamaterial mechanisms [IFW+16, ILH+19], is to determine the stiffness data in equations of elasticity from displacement and boundary data. Several variants of this problem are exciting: one could try to find solutions in the most general stiffness space, where all thermodynamically admissible materials, i.e., positive-definite stiffness tensors, are allowed, and fabricate them using dehomogenization techniques. One could also restrict the search space to spatially-varying isotropic materials, i.e., find solutions to the Navier–Cauchy equations, and fabricate them on 3d printers allowing for materials with continuously-varying stiffness.

There are also several subproblems depending on how much of the displacement data is described. Characterizing all displacement fields that admit a solution would provide theoretical insight into the physical limits of elasticity, but finding solutions in which only displacements on part of the boundary are fixed may be most useful for practical applications such as the discovery of elastic mechanisms. Another interesting question which is relevant to mechanism design is whether the inverse problem becomes fundamentally more difficult if non-linear equations of elasticity, such as those that arise from applying the St. Venant–Kirchhoff material model, are used. I suspect that this is not the case because from an "inverse point of view", the unknowns and their constraints do not change if non-linearity is introduced in the strain computation.

I hope that I have managed to transport some of my excitement about studying inverse problems in fabrication-aware optimization to the reader. It is my strong belief that there are many beautiful and useful results waiting to be discovered here.

Bibliography

[AP10] B. Audoly and Y. Pomeau. *Elasticity and Geometry: From Hair Curls to the Non-linear Response of Shells*. Oxford University Press, 2010.

[AV93] K Atkinson and Ezio Venturino. Numerical evaluation of line integrals. *SIAM journal on numerical analysis*, 30(3):882–888, 1993.

[BAC+06] Florence Bertails, Basile Audoly, Marie-Paule Cani, Bernard Querleux, Frédéric Leroy, and Jean-Luc Lévêque. Super-helices for predicting the dynamics of natural hair. *ACM Transactions on Graphics*, 25(3):1180–1187, 2006.

[Bat15] Milan Batista. On stability of elastic rod planar equilibrium configurations. *International Journal of Solids and Structures*, 72:144 – 152, 2015.

[BAV+10] Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. Discrete viscous threads. *ACM Transactions on Graphics*, 29, 07 2010.

[BB99] T. Belytschko and T. Black. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45(5):601–620, jun 1999.

[BBO+10] Bernd Bickel, Moritz Bächer, Miguel A. Otaduy, Hyunho Richard Lee, Hanspeter Pfister, Markus Gross, and Wojciech Matusik. Design and fabrication of materials with desired deformation behavior. *ACM Trans. Graph.*, 29(4), jul 2010.

[BCMP18] Bernd Bickel, Paolo Cignoni, Luigi Malomo, and Nico Pietroni. State of the art on stylized fabrication. *Computer Graphics Forum*, 37(6):325–342, 2018.

[BDJRL18] Florence Bertails-Descoubes, Alexandre Jourdan, Victor Romero, and Arnaud Lazarus. Inverse design of an isotropic suspended kirchhoff rod: theoretical and numerical results on the uniqueness of the natural shape. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2212):20170837, 2018.

[BFR17] Amit H. Bermano, Thomas Funkhouser, and Szymon Rusinkiewicz. State of the art in methods and representations for fabrication-aware design. *Computer Graphics Forum*, 36(2):509–535, 2017.

[BHLH88] Chandrajit L Bajaj, Christoph M Hoffmann, Robert E Lynch, and JEH Hopcroft. Tracing surface intersections. *Computer aided geometric design*, 5(4):285–307, 1988.

[Bis75] Richard L. Bishop. There is more than one way to frame a curve. *The American Mathematical Monthly*, 82(3):246–251, 1975.

[BL10] J.Y. Bello Cruz and L.R. Lucambio Pérez. Convergence of a projected gradient method variant for quasiconvex objectives. *Nonlinear Analysis: Theory, Methods & Applications*, 73(9):2917–2922, 2010.

[BLT+15] Gaurav Bharaj, David I. W. Levin, James Tompkin, Yun Fei, Hanspeter Pfister, Wojciech Matusik, and Changxi Zheng. Computational design of metallophone contact sounds. *ACM Transactions on Graphics*, 34(6), 2015.

[Bol02] Oskar Bolza. Proof of the sufficiency of jacobi's condition for a permanent sign of the second variation in the so-called isoperimetric problems. *Transactions of the American Mathematical Society*, 3(3):305–311, 1902.

[Bol03] Oskar Bolza. Zur zweiten variation bei isoperimetrischen problemen. *Mathematische Annalen*, 56:44–47, 1903.

[Bol04] Oskar Bolza. *Lectures on the Calculus of Variations*. Chelsea Publishing Company, 1904.

[Bor06] Max Born. *Untersuchungen über die Stabilität der elastische Linie in Ebene und Raum, unter verschiedenen Grenzbedingungen*. Göttingen, Dieterich'sche Univ.-Buchdruckerei, 1906.

[Bra07] Dietrich Braess. *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics*. Cambridge University Press, Cambridge, England, 3 edition, 2007.

[Bra19] Andrew M. Bradley. Pde-constrained optimization and the adjoint method. `https://cs.stanford.edu/~ambrad/adjoint_tutorial.pdf`, 2019.

[BS99] M. P. Bendsøe and O. Sigmund. Material interpolation schemes in topology optimization. *Archive of Applied Mechanics (Ingenieur Archiv)*, 69(9-10):635–654, nov 1999.

[BWBSH14] Moritz Bächer, Emily Whiting, Bernd Bickel, and Olga Sorkine-Hornung. Spin-it. *ACM Trans. Graph.*, 33(4), 2014.

[BWR+08] Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. Discrete elastic rods. In *ACM Transactions on Graphics*. ACM Press, 2008.

[CCC16] Jorge D. Camba, Manuel Contero, and Pedro Company. Parametric cad modeling: An analysis of strategies for design reusability. *Computer-Aided Design*, 74:18–31, 2016.

[Cea86] Jean Cea. Conception optimale ou identification de formes, calcul rapide de la dérivée directionnelle de la fonction coût. *Mathematical Modelling and Numerical Analysis*, 20:371–402, 1986.

[CH14] Jen-San Chen and Shao-Yu Hung. Deformation and stability of an elastica constrained by curved surfaces. *International Journal of Mechanical Sciences*, 82:1–12, 2014.

[CHB09] J. Austin Cottrell, Thomas J. R Hughes, and Yuri Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, 2009.

[CLPS03] Yang Cao, Shengtai Li, Linda Petzold, and Radu Serban. Adjoint sensitivity analysis for differential-algebraic equations: The adjoint dae system and its numerical solution. *SIAM Journal on Scientific Computing*, 24:1076–1089, 01 2003.

[CMM18] Pierre Cuvilliers, Paul Mayencourt, and Caitlin Mueller. The arc lamp workshop at aag 2018: active bending and digital fabrication, 2018. Accessed: 2021-01-27.

[Coş10] S.B. Coşkun. Analysis of tilt-buckling of euler columns with varying flexural stiffness using homotopy perturbation method. *Mathematical Modelling and Analysis*, 15(3):275–286, 2010.

[CZXZ14] Xiang Chen, Changxi Zheng, Weiwei Xu, and Kun Zhou. An asymptotic numerical method for inverse elastic shape design. *ACM Transactions on Graphics*, 33(4), jul 2014.

[DGJ18] Fernando De Goes and Doug L. James. Dynamic kelvinlets: Secondary motions based on fundamental solutions of elastodynamics. *ACM Trans. Graph.*, 37(4), jul 2018.

[DHL14] Jérémie Dumas, Jean Hergel, and Sylvain Lefebvre. Bridging the gap: Automated steady scaffoldings for 3d printing. *ACM Trans. Graph.*, 33(4), jul 2014.

[DJBDDT13] Alexandre Derouet-Jourdan, Florence Bertails-Descoubes, Gilles Daviet, and Joëlle Thollot. Inverse dynamic hair modeling with frictional contact. *ACM Transactions on Graphics*, 32(6):1–10, 2013.

[DJBDT10] Alexandre Derouet-Jourdan, Florence Bertails-Descoubes, and Joëlle Thollot. Stable inverse dynamic curves. *ACM Transactions on Graphics*, 29, 2010.

[DLL+15] Jérémie Dumas, An Lu, Sylvain Lefebvre, Jun Wu, and Christian Dick. By-example synthesis of structurally sound patterns. *ACM Trans. Graph.*, 34(4), jul 2015.

[DMJF06] Pierre Duysinx, Laurent Van Miegroet, Thibault Jacobs, and Claude Fleury. Generalized shape optimization using x-FEM and level set methods. In *Solid Mech. and Its App.*, pages 23–32. Springer Netherlands, 2006.

[DP17] Sachin D. Daxini and Jagdish M. Prajapati. Parametric shape optimization techniques based on meshless methods: a review. *Structural and Multidisciplinary Optimization*, 56(5):1197–1214, may 2017.

[DPTC20] Simon Duenser, Roi Poranne, Bernhard Thomaszewski, and Stelian Coros. Robocut: Hot-wire cutting with robot-controlled flexible rods. *ACM Transactions on Graphics*, 39(4), jul 2020.

[DSMA+21] H. Djavaherpour, F. Samavati, A. Mahdavi-Amiri, F. Yazdanbakhsh, S. Huron, R. Levy, Y. Jansen, and L. Oehlberg. Data to physicalization: A survey of the physical rendering process. *Computer Graphics Forum*, 40(3):569–598, 2021.

[DVTM16] Levi H Dudte, Etienne Vouga, Tomohiro Tachi, and Lakshminarayanan Mahadevan. Programming curvature using origami tessellations. *Nature materials*, 15(5):583–588, 2016.

[DW48] J. B. Diaz and A. Weinstein. The torsional rigidity and variational methods. *American Journal of Mathematics*, 70(1):107–116, 1948.

[EKS+10] Michael Eigensatz, Martin Kilian, Alexander Schiftner, Niloy J. Mitra, Helmut Pottmann, and Mark Pauly. Paneling architectural freeform surfaces. *ACM Trans. Graph.*, 29(4), jul 2010.

[ESZ+17] Oskar Elek, Denis Sumin, Ran Zhang, Tim Weyrich, Karol Myszkowski, Bernd Bickel, Alexander Wilkie, and Jaroslav Křivánek. Scattering-aware texture reproduction for 3d printing. *ACM Trans. Graph.*, 36(6), nov 2017.

[FB10] Thomas-Peter Fries and Ted Belytschko. The extended/generalized finite element method: An overview of the method and its applications. *International Journal for Numerical Methods in Engineering*, aug 2010.

[FGBP11] François Faure, Benjamin Gilles, Guillaume Bousquet, and Dinesh K. Pai. Sparse meshless models of complex deformable solids. *ACM Trans. Graph.*, 30(4):1, 2011.

[FM11] Moritz Fleischmann and Achim Menges. ICD/ITKE research pavilion: A case study of multi-disciplinary collaborative computational design. In *Computational Design Modelling*, pages 239–248. Springer Berlin Heidelberg, 2011.

[FMT95] Gilles Francfort, François Murat, and Luc Tartar. Fourth-order moments of nonnegative measures on s2 and applications. *Archive for Rational Mechanics and Analysis*, 131:305–333, 12 1995.

[GF63] I. M. Gelfand and S. V. Fomin. *Calculus of Variations*. Prentice Hall, 1963.

[GGP+20] 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 Transactions on Graphics*, 39(6), 2020.

[GMB17] Ruslan Guseinov, Eder Miguel, and Bernd Bickel. CurveUps. *ACM Transactions on Graphics*, 36(4):1–12, 2017.

[GMP+20] 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.

[Gol80] Herman H. Goldstine. *A History of the Calculus of Variations from the 17th through the 19th Century*. Springer-Verlag New York, 1980.

[GSFD+14] Akash Garg, Andrew O. Sageman-Furnas, Bailin Deng, Yonghao Yue, Eitan Grinspun, Mark Pauly, and Max Wardetzky. Wire mesh design. *ACM Transactions on Graphics*, 33(4):1–12, 2014.

[Had06] Sunil Hadap. Oriented strands: Dynamics of stiff multi-body system, 2006.

[HB21] Christian Hafner and Bernd Bickel. The design space of plane elastic curves. *ACM Trans. Graph.*, 40(4), jul 2021.

[HB22] Christian Hafner and Bernd Bickel. The design space of kirchhoff rods. *Accepted to ACM Trans. Graph. with minor revision*, nov 2022.

[Hes66] M.R. Hestenes. *Calculus of Variations and Optimal Control Theory*. Applied mathematics series. Wiley, 1966.

[HJ19] Udo Hertrich-Jeromin. Differential geometry. `https://www.geometrie.tuwien.ac.at/hertrich-jeromin/pub/dg.pdf`, 2019. Lecture Notes.

[HKPU11] G. Haasemann, M. Kästner, S. Prüger, and V. Ulbricht. Development of a quadratic finite element formulation based on the XFEM and NURBS. *International Journal for Numerical Methods in Engineering*, 86(4-5):598–617, jan 2011.

[Hof05] Christoph Hoffmann. Constraint-based computer-aided design. *Journal of Computing and Information Science in Engineering*, 5:182, 09 2005.

[HPL+19] Liang He, Huaishu Peng, Michelle Lin, Ravikanth Konjeti, François Guimbretière, and Jon E. Froehlich. Ondulé: Designing and controlling 3d printable springs. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. ACM, 2019.

[HSK+19] Christian Hafner, Christian Schumacher, Espen Knoop, Thomas Auzinger, Bernd Bickel, and Moritz Bächer. X-cad: Optimizing cad models with extended finite elements. *ACM Trans. Graph.*, 38(6), nov 2019.

[HWH+15] Ming-Chen Hsu, Chenglong Wang, Austin J. Herrema, Dominik Schillinger, Anindya Ghoshal, and Yuri Bazilevs. An interactive geometry modeling and parametric design platform for isogeometric analysis. *Computers & Mathematics with Applications*, 70(7):1481–1500, oct 2015.

[HXK+19] Shayan Hoshyari, Hongyi Xu, Espen Knoop, Stelian Coros, and Moritz Bächer. Vibration-minimizing motion retargeting for robotic characters. *ACM Transactions on Graphics*, 38(4), jul 2019.

[HZG+18] Yixin Hu, Qingnan Zhou, Xifeng Gao, Alec Jacobson, Denis Zorin, and Daniele Panozzo. Tetrahedral meshing in the wild. *ACM Trans. Graph.*, 37(4):60:1–60:14, July 2018.

[IFW+16] Alexandra Ion, Johannes Frohnhofen, Ludwig Wall, Robert Kovacs, Mirela Alistar, Jack Lindsay, Pedro Lopes, Hsiang-Ting Chen, and Patrick Baudisch. Metamaterial mechanisms. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, page 529–539, New York, NY, USA, 2016. Association for Computing Machinery.

[ILH+19] Alexandra Ion, David Lindlbauer, Philipp Herholz, Marc Alexa, and Patrick Baudisch. Understanding metamaterial mechanisms. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–14, New York, NY, USA, 2019. Association for Computing Machinery.

[IRHSH20] Alexandra Ion, Michael Rabinovich, Philipp Herholz, and Olga Sorkine-Hornung. Shape approximation by developable wrapping. *ACM Transactions on Graphics*, 39(6):1–12, 2020.

[JK09] Lenka Jeřábková and Torsten Kuhlen. Stable cutting of deformable objects in virtual environments using XFEM. *IEEE Computer Graphics and Applications*, 29(2):61–71, mar 2009.

[JSVB20] David Jourdan, Melina Skouras, Etienne Vouga, and Adrien Bousseau. Printing-on-fabric meta-material for self-shaping architectural models. In *Advances in Architectural Geometry*, 2020.

[KBT17] Dan Koschier, Jan Bender, and Nils Thuerey. Robust extended finite elements for complex cutting of deformables. *ACM Trans. Graph.*, 36(4):55:1–55:13, July 2017.

[KCD+16] Mina Konaković, Keenan Crane, Bailin Deng, Sofien Bouaziz, Daniel Piker, and Mark Pauly. Beyond developable: Computational design and fabrication with auxetic materials. *ACM Trans. Graph.*, 35(4), jul 2016.

[Kir11] Andreas Kirsch. *An Introduction to the Mathematical Theory of Inverse Problems*, volume 120. Springer New York, 01 2011.

[KLPCP18] Mina Konaković-Luković, Julian Panetta, Keenan Crane, and Mark Pauly. Rapid deployment of curved surfaces via programmable auxetics. *ACM Transactions on Graphics*, 37(4):1–13, 2018.

[KMB+09] Peter Kaufmann, Sebastian Martin, Mario Botsch, Eitan Grinspun, and Markus Gross. Enrichment textures for detailed cutting of shells. *ACM Trans. Graph.*, 28(3), 2009.

[KMJ+19] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[Kne02] Adolf Kneser. Beiträge zur theorie und anwendung der variationsrechnung (erster aufsatz). *Mathematische Annalen*, 55:86–107, 1902.

[Kot14] M. Kot. *A First Course in the Calculus of Variations*. Student Mathematical Library. American Mathematical Society, 2014.

[Kov69] K. Kovari. Räumliche verzweigungsprobleme des dünnen elastischen stabes mit endlichen verformungen. *Ingenieur Archiv*, 37:393–416, 1969.

[KSOJ12] 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 IASS, editor, *Proceedings of the IASS-APCS-Symposium*, Seoul, Korea, 2012.

[KZB+15] László Kudela, Nils Zander, Tino Bog, Stefan Kollmannsberger, and Ernst Rank. Efficient and accurate numerical quadrature for immersed boundary methods. *Advanced Modeling and Simulation in Engineering Sciences*, 2(1):10, 2015.

[LAGK13] 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.

[LDV20] Mingchao Liu, Lucie Domino, and Dominic Vella. Tapered elasticæ as a route for axisymmetric morphing structures. *Soft Matter*, 16:7739–7750, 2020.

[Leg13] Grégory Legrain. A NURBS enhanced extended finite element approach for unfitted CAD analysis. *Computational Mechanics*, 52(4):913–929, apr 2013.

[Lev08] Raph Levien. The elastica: a mathematical history. Technical report, Electrical Engineering and Computer Sciences, University of California at Berkeley, 2008.

[LFL09] Kui-Yip Lo, Chi-Wing Fu, and Hongwei Li. 3d polyomino puzzle. *ACM Trans. Graph.*, 28(5):1–8, dec 2009.

[LHZ+18] Haixiang Liu, Yuanming Hu, Bo Zhu, Wojciech Matusik, and Eftychios Sifakis. Narrow-band topology optimization on a sparsely populated grid. *ACM Trans. Graph.*, 37(6):251:1–251:14, December 2018.

[Liu16] G. R. Liu. On partitions of unity property of nodal shape functions: Rigid-body-movement reproduction and mass conservation. *International Journal of Computational Methods*, 13(02):1640003, 2016.

[LJGH11] Xian-Ying Li, Tao Ju, Yan Gu, and Shi-Min Hu. A geometric study of v-style pop-ups: Theories and algorithms. *ACM Trans. Graph.*, 30(4), jul 2011.

[LL01] E.H. Lieb and M. Loss. *Analysis*. Crm Proceedings & Lecture Notes. American Mathematical Society, 2001.

[LL18] Joon Kyu Lee and Byoung Koo Lee. Elastica and buckling loads of nonlinear elastic tapered cantilever columns. *Engineering Solid Mechanics*, 6:39–50, 2018.

[LLMZ16] Dingzeyu Li, David I. W. Levin, Wojciech Matusik, and Changxi Zheng. Acoustic voxels: Computational optimization of modular acoustic filters. *ACM Trans. Graph.*, 35(4), jul 2016.

[MAB+15] Przemyslaw Musialski, Thomas Auzinger, Michael Birsak, Michael Wimmer, and Leif Kobbelt. Reduced-order shape optimization using offset surfaces. *ACM Trans. Graph.*, 34(4):102:1–102:9, 2015.

[Mad81] J. H. Maddocks. *Analysis of nonlinear differential equations governing the equilibria of an elastic rod and their stability*. PhD thesis, University of Oxford, 1981.

[MBA+18] Orest Mykhaskiv, Mladen Banovic, Salvatore Auriemma, Pavanakumar Mohanamuraly, Andrea Walther, Herve Legrand, and Jens-Dominik Müller. Nurbs-based and parametric-based shape optimization with differentiated cad kernel. *Computer-Aided Design and Applications*, 15(6):916–926, 2018.

[MHR+16] Przemyslaw Musialski, Christian Hafner, Florian Rist, Michael Birsak, Michael Wimmer, and Leif Kobbelt. Non-linear shape optimization using local subspace projections. *ACM Trans. Graph.*, 35(4), 2016.

[MIB15] Masaaki Miki, Takeo Igarashi, and Philippe Block. Parametric self-supporting surfaces via direct computation of airy stress functions. *ACM Trans. Graph.*, 34(4), jul 2015.

[Mie15] Erich Miersemann. Calculus of variations. `http://www.math.uni-leipzig.de/~miersemann/variations2015.pdf`, 2015. Lecture Notes.

[MKB+10] Sebastian Martin, Peter Kaufmann, Mario Botsch, Eitan Grinspun, and Markus Gross. Unified simulation of elastic rods, shells, and solids. *ACM Trans. Graph.*, 29(4):1, 2010.

[MKEKO17] B. Müller, S. Krämer-Eis, F. Kummer, and M. Oberlack. A high-order discontinuous galerkin method for compressible flows with immersed boundaries. *International Journal for Numerical Methods in Engineering*, 110(1):3–30, 2017.

[MKN+04] M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, and M. Alexa. Point based animation of elastic, plastic and melting objects. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation - SCA '04*. ACM Press, 2004.

[MKO13] B Müller, F Kummer, and M Oberlack. Highly accurate surface and volume integration on implicit domains by means of moment-fitting. *International Journal for Numerical Methods in Engineering*, 96(8):512–528, 2013.

[MLB16] Eder Miguel, Mathias Lepoutre, and Bernd Bickel. Computational design of stable planar-rod structures. *ACM Transactions on Graphics*, 35(4):1–11, 2016.

[MPI+19] Luigi Malomo, Jesús Pérez, Emmanuel Iarussi, Nico Pietroni, Eder Miguel, Paolo Cignoni, and Bernd Bickel. FlexMaps. *ACM Transactions on Graphics*, 37(6):1–14, 2019.

[MRA88] Jerrold E. Marsden, Tudor Ratiu, and Ralph Abraham. *Manifolds, Tensor Analysis, and Applications*. Springer, New York, 1988.

[MRM98] Robert S. Manning, Kathleen A. Rogers, and John H. Maddocks. Isoperimetric conjugate points with application to the stability of dna minicircles. *Proceedings: Mathematical, Physical and Engineering Sciences*, 454(1980):3047–3074, 1998.

[Nit71] J. Nitsche. Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36(1):9–15, Jul 1971.

[NSTG17] Ahmad R. Najafi, Masoud Safdari, Daniel A. Tortorelli, and Philippe H. Geubelle. Shape optimization using a NURBS-based interface-enriched generalized FEM. *International Journal for Numerical Methods in Engineering*, 111(10):927–954, jan 2017.

[NTV92] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics*, 10(5):307–318, 1992.

[NW06] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006.

[O'N66] Barrett O'Neill. *Elementary Differential Geometry*. Academic Press, New York, 1966.

[Pai02] Dinesh K. Pai. STRANDS: Interactive simulation of thin solids using cosserat models. *Computer Graphics Forum*, 21(3):347–352, 2002.

[PIC+21] Julian Panetta, Florin Isvoranu, Tian Chen, Emmanuel Siéfert, Benoît Roman, and Mark Pauly. Computational inverse design of surface-based inflatables. *ACM Transactions on Graphics*, 40(4), jul 2021.

[PKLI+19] J. Panetta, M. Konaković-Luković, F. Isvoranu, E. Bouleau, and M. Pauly. X-shells: A new class of deployable beam structures. *ACM Transactions on Graphics*, 38(4):1–15, 2019.

[PKPP21] Davide Pellis, Martin Kilian, Helmut Pottmann, and Mark Pauly. Computational design of weingarten surfaces. *ACM Trans. Graph.*, 40(4), jul 2021.

[PLBM20] Stefan Pillwein, Kurt Leimer, Michael Birsak, and Przemyslaw Musialski. On elastic geodesic grids and their planar to spatial deployment. *ACM Transactions on Graphics*, 39(4), 2020.

[PM21] Stefan Pillwein and Przemyslaw Musialski. Generalized deployable elastic geodesic grids. *ACM Transactions on Graphics*, 40(6), dec 2021.

[POT17] Jesús Pérez, Miguel A. Otaduy, and Bernhard Thomaszewski. Computational design and automated fabrication of kirchhoff-plateau surfaces. *ACM Transactions on Graphics*, 36(4):1–12, 2017.

[PT96] Les Piegl and Wayne Tiller. *The NURBS Book*. Springer-Verlag, New York, NY, USA, second edition, 1996.

[PTC+15] Jesús Pérez, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, José A. Canabal, Robert Sumner, and Miguel A. Otaduy. Design and fabrication of flexible rod meshes. *ACM Transactions on Graphics*, 34(4):1–12, 2015.

[PW01] Helmut Pottmann and Johannes Wallner. *Computational Line Geometry*. Springer-Verlag, Berlin, Heidelberg, 2001.

[PZM+15] Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. Elastic textures for additive fabrication. *ACM Trans. Graph.*, 34(4), jul 2015.

[RAM+21] Marie-Julie Rakotosaona, Noam Aigerman, Niloy J. Mitra, Maks Ovsjanikov, and Paul Guerrero. Differentiable surface triangulation. *ACM Trans. Graph.*, 40(6), dec 2021.

[RDM19] Mark Runciman, Ara Darzi, and George P. Mylonas. Soft robotics in minimally invasive surgery. *Soft Robotics*, 6(4):423–443, 2019. PMID: 30920355.

[Ric81] Gerard R. Richter. An inverse problem for the steady state diffusion equation. *SIAM Journal on Applied Mathematics*, 41(2):210–221, 1981.

[Rol91] D. Roller. An approach to computer-aided parametric design. *Computer-Aided Design*, 23(5):385–391, 1991.

[RPC+21] 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*, 40(4), jul 2021.

[SB12] Eftychios Sifakis and Jernej Barbič. Fem simulation of 3d deformable solids: A practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses*, SIGGRAPH '12, pages 20:1–20:50, New York, NY, USA, 2012. ACM.

[SDW+16] Peng Song, Bailin Deng, Ziqi Wang, Zhichao Dong, Wei Li, Chi-Wing Fu, and Ligang Liu. Cofifab: Coarse-to-fine fabrication of large 3d objects. *ACM Trans. Graph.*, 35(4), jul 2016.

[SFCBCV19] Andrew O. Sageman-Furnas, Albert Chern, Mirela Ben-Chen, and Amir Vaxman. Chebyshev nets from commuting polyvector fields. *ACM Transactions on Graphics*, 38(6), nov 2019.

[SGC18] Oded Stein, Eitan Grinspun, and Keenan Crane. Developability of triangle meshes. *ACM Transactions on Graphics*, 37(4):1–14, 2018.

[Sha91] J.J. Shah. Assessment of features technology. *Computer-Aided Design*, 23(5):331–343, 1991.

[Sha18] Nicholas Sharp. Céa's method for pde-constrained shape optimization. `https://nmwsharp.com/media/cea_tutorial.pdf`, 2018.

[SHD+18] Teseo Schneider, Yixin Hu, Jérémie Dumas, Xifeng Gao, Daniele Panozzo, and Denis Zorin. Decoupling simulation accuracy from mesh quality. *ACM Trans. Graph.*, 37(6), December 2018.

[SHSP18] Eike Schling, Wang Hui, Jonas Schikore, and Helmut Pottmann. Design and construction of curved support structures with repetitive parameters. In *Advances in Architectural Geometry 2018*, pages 140–165, Vienna, Austria, 09 2018. Klein Publishing.

[SL10] Yuri Sachkov and Stanislav Levyakov. Stability of inflectional elasticae centered at vertices or inflection points. *Proceedings of the Steklov Institute of Mathematics*, 271:177–192, 2010.

[SNSG15] Masoud Safdari, Ahmad R. Najafi, Nancy R. Sottos, and Philippe H. Geubelle. A NURBS-based interface-enriched generalized finite element method for problems with complex discontinuous gradient fields. *International Journal for Numerical Methods in Engineering*, 101(12):950–964, jan 2015.

[SNSG16] Masoud Safdari, Ahmad R Najafi, Nancy R Sottos, and Philippe H Geubelle. A nurbs-based generalized finite element scheme for 3d simulation of heterogeneous materials. *Journal of Computational Physics*, 318:373–390, 2016.

[SOG+22] Florian Cyril Stutz, Tim Felle Olsen, Jeroen Peter Groen, Tuan Nguyen Trung, Niels Aage, Ole Sigmund, Justin Solomon, and Jakob Andreas Bærentzen. Synthesis of frame field-aligned multi-laminar structures. *ACM Trans. Graph.*, 41(5), may 2022.

[SSB13] F. S. Sin, D. Schroeder, and J. Barbič. Vega: Non-linear fem deformable object simulator. *Computer Graphics Forum*, 32(1):36–48, 2013.

[STBG12] Mélina Skouras, Bernhard Thomaszewski, Bernd Bickel, and Markus Gross. Computational design of rubber balloons. *Comput. Graph. Forum*, 31(2pt4):835–844, May 2012.

[STC+13] Mélina Skouras, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, and Markus Gross. Computational design of actuated deformable characters. *ACM Trans. Graph.*, 32(4):82:1–82:10, July 2013.

[Sut63] Ivan E. Sutherland. Sketchpad: A man-machine graphical communication system. In *Proceedings of the May 21-23, 1963, Spring Joint Computer Conference*, AFIPS '63 (Spring), pages 329–346, New York, NY, USA, 1963. ACM.

[SVB+12] Ondrej Stava, Juraj Vanek, Bedrich Benes, Nathan Carr, and Radomír Měch. Stress relief: Improving structural strength of 3d printable objects. *ACM Trans. Graph.*, 31(4):48:1–48:11, July 2012.

[SXZ+17] Adriana Schulz, Jie Xu, Bo Zhu, Changxi Zheng, Eitan Grinspun, and Wojciech Matusik. Interactive design space exploration and optimization for cad models. *ACM Trans. Graph.*, 36(4):157:1–157:14, July 2017.

[Sym15] Ioanna Symeonidou. Analogue and digital form-finding of bending rod structures. In *Proceedings of IASS 2015 Amsterdam Symposium: Future Visions – Computational Design*, pages 1–12. International Association for Shell and Spatial Structures (IASS), 2015.

[SZB18] Christian Schumacher, Jonas Zehnder, and Moritz Bächer. Set-in-stone: Worst-case optimization of structures weak in tension. *ACM Trans. Graph.*, 37(6):252:1–252:13, December 2018.

[TBWP16] Chengcheng Tang, Pengbo Bo, Johannes Wallner, and Helmut Pottmann. Interactive design of developable surfaces. *ACM Trans. Graph.*, 35(2), jan 2016.

[UKSI14] Nobuyuki Umetani, Yuki Koyama, Ryan Schmidt, and Takeo Igarashi. Pteromys: Interactive design and optimization of free-formed free-flight model airplanes. *ACM Trans. Graph.*, 33(4), jul 2014.

[VdAMM07] Nico Van der Aa, H.G. Morsche, and R.R.M. Mattheij. Computation of eigenvalue and eigenvector derivatives for a general complex-valued eigensystem. *The Electronic Journal of Linear Algebra*, 16, 2007.

[VHWP12] Etienne Vouga, Mathias Höbinger, Johannes Wallner, and Helmut Pottmann. Design of self-supporting surfaces. *ACM Trans. Graph.*, 31(4), jul 2012.

[VWRKM13] Kiril Vidimče, Szu-Po Wang, Jonathan Ragan-Kelley, and Wojciech Matusik. Openfab: A programmable pipeline for multi-material fabrication. *ACM Transactions on Graphics*, 32, July 2013.

[VZF+19] Josh Vekhter, Jiacheng Zhuo, Luisa F. Gil Fandino, Qixing Huang, and Etienne Vouga. Weaving geodesic foliations. *ACM Transactions on Graphics*, 38(4), jul 2019.

[WPGSH18] Katja Wolff, Roi Poranne, Oliver Glauser, and Olga Sorkine-Hornung. Packable springs. *Computer Graphics Forum*, 37(2):251–262, 2018.

[WS21] Yu Wang and Justin Solomon. Fast quasi-harmonic weights for geometric data interpolation. *ACM Transactions on Graphics*, 40(4), jul 2021.

[WSP21] Ziqi Wang, Peng Song, and Mark Pauly. State of the art on computational design of assemblies with rigid parts. *Computer Graphics Forum*, 40(2):633–657, 2021.

[WTC+19] Guanyun Wang, Ye Tao, Ozguc Bertug Capunaman, Humphrey Yang, and Lining Yao. A-line: 4d printing morphing linear composite structures. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, may 2019.

[WWY+13] Weiming Wang, Tuanfeng Y. Wang, Zhouwang Yang, Ligang Liu, Xin Tong, Weihua Tong, Jiansong Deng, Falai Chen, and Xiuping Liu. Cost-effective printing of 3d objects with skin-frame structures. *ACM Trans. Graph.*, 32(6), nov 2013.

[XKCB19] Hongyi Xu, Espen Knoop, Stelian Coros, and Moritz Bächer. Bend-it: Design and fabrication of kinetic wire characters. *ACM Transactions on Graphics*, 37(6):1–15, 2019.

[ZCT16] Jonas Zehnder, Stelian Coros, and Bernhard Thomaszewski. Designing structurally-sound ornamental curve networks. *ACM Transactions on Graphics*, 35(4), jul 2016.

[ZFF+19] Xu Zheng, Zhichao Fan, Haoran Fu, Yuan Liu, Yanyang Zi, Yonggang Huang, and Yihui Zhang. Optimization-based approach for the inverse design of ribbon-shaped three-dimensional structures assembled through compressive buckling. *Physical Review Applied*, 11, 2019.

[ZGH+16] Haisen Zhao, Fanglin Gu, Qi-Xing Huang, Jorge Garcia, Yong Chen, Changhe Tu, Bedrich Benes, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. Connected fermat spirals for layered fabrication. *ACM Trans. Graph.*, 35(4), jul 2016.

[ZKBT17] Jonas Zehnder, Espen Knoop, Moritz Bächer, and Bernhard Thomaszewski. Metasilicone. *ACM Trans. Graph.*, 36(6), 2017.

[ZKWG16] Yahan Zhou, Evangelos Kalogerakis, Rui Wang, and Ian R. Grosse. Direct shape optimization for strengthening 3d printable objects. *Computer Graphics Forum*, 35(7):333–342, oct 2016.

[Zou70] G. Zoutendijk. Computational methods in nonlinear programming. *Studies in Optimization*, 1:125, 1970.

[ZPZ13] Qingnan Zhou, Julian Panetta, and Denis Zorin. Worst-case structural analysis. *ACM Trans. Graph.*, 32(4), jul 2013.

APPENDIX A

Plane Elastic Curves

A.1 Proof of Geometric Characterization

Here we give a proof of Theorem 1. It uses a short lemma, which we show first.

Lemma 11. *With the notation from Theorem 1, let $s_0 \in S_0$. Furthermore, let (t, n) be the right-handed orthonormal basis adapted to γ at s_0, i.e., $t = \gamma'(s_0)$. Then, (1) there exists a neighborhood of s_0, in which s_0 is the only root of $s \mapsto \langle \gamma''(s), n\rangle$; (2) it holds that*

$$\lim_{s\to s_0} \frac{\kappa(s)}{\langle \gamma''(s), n\rangle} = 1.$$

Proof. (1) We can find $\varepsilon > 0$ small enough, such that for all $s \in S_\varepsilon := (s_0-\varepsilon, s_0)\cup(s_0, s_0+\varepsilon)$, we have $\kappa(s) \neq 0$ and $\langle \gamma'(s), t\rangle > \frac{1}{2}$. Assume for the sake of contradiction that there exists $s_1 \in S_\varepsilon$ with $\langle \gamma''(s_1), n\rangle = 0$. Then, it follows from $0 \neq |\kappa(s_1)| = \|\gamma''(s_1)\|$ that $\langle \gamma''(s_1), t\rangle \neq 0$. This lets us compute

$$\langle \gamma'(s_1), t\rangle = \frac{1}{\langle \gamma''(s_1), t\rangle}\langle \gamma'(s_1), \gamma''(s_1)\rangle = 0,$$

because $\langle \gamma', \gamma''\rangle \equiv 0$. But this contradicts $\langle \gamma'(s_1), t\rangle > \frac{1}{2}$.

(2) By expressing κ in the coordinate system (t, n), we get

$$\frac{\kappa}{\langle \gamma'', n\rangle} = \frac{\det(\gamma', \gamma'')}{\langle \gamma'', n\rangle} = \langle \gamma', t\rangle - \frac{\langle \gamma', n\rangle\langle \gamma'', t\rangle}{\langle \gamma'', n\rangle}.$$

For the first term on the right-hand side, we have $\lim_{s\to s_0}\langle \gamma'(s), t\rangle = 1$. For the second term, write $\langle \gamma', \gamma''\rangle \equiv 0$ in coordinates to get

$$\langle \gamma', n\rangle\langle \gamma'', n\rangle + \langle \gamma', t\rangle\langle \gamma'', t\rangle = 0, \text{ so } \langle \gamma'', t\rangle = -\frac{\langle \gamma', n\rangle\langle \gamma'', n\rangle}{\langle \gamma', t\rangle},$$

where the denominator is non-zero in S_ε. Using this equality, the second term becomes

$$-\frac{\langle \gamma', n\rangle\langle \gamma'', t\rangle}{\langle \gamma'', n\rangle} = \frac{\langle \gamma', n\rangle^2}{\langle \gamma', t\rangle},$$

which goes to zero as $s \to s_0$, because $\langle \gamma'(s_0), n\rangle = 0$. This shows the statement. □

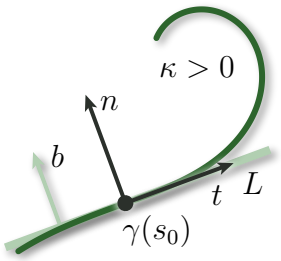
Proof of Theorem 1. **Direction "⇒":** Assume γ is an equilibrium curve, so there exist $a \in \mathbb{R}$, $b \in \mathbb{R}^2$, and an admissible stiffness function K with $K\kappa = a + \langle b, \gamma \rangle$. We show that then (1) and (2) hold. From $K > 0$, it follows that either $a \neq 0$ or $b \neq 0$, because otherwise $\kappa \equiv 0$, which contradicts that S_0 is finite.

Case $b = 0$: Then, $a \neq 0$, and κ has the same sign as a everywhere, so there are no zero-curvature points, and (2) is vacuously true. Choose any line L that does not intersect γ to satisfy (1).

Case $b \neq 0$**: Proof of (1):** The level sets of the function $f : \mathbb{R}^2 \to \mathbb{R} : x \mapsto a + \langle b, x \rangle$ form a family of parallel lines. Let L be the zero-level set of f. We have $0 = K(s)\kappa(s)$ if and only if $s \in S_0$, and thus $0 = a + \langle b, \gamma(s) \rangle$ if and only if $s \in S_0$. This shows that γ and L intersect exactly in $\gamma(S_0)$.

Next, we show that L is not tangent to γ in any of the zero-curvature points. With the notation from Lemma 11, assume for the sake of contradiction that L is tangent to γ at s_0, i.e., $\langle b, t \rangle = 0$, which implies $b = \langle b, n \rangle n$.

⌈**Claim:** $\langle b, n \rangle > 0$. **Proof of claim:** By S_0 finite, we know that κ is non-zero in $(s_0, s_0 + \varepsilon)$ for $\varepsilon > 0$ small enough, so it does not change sign in this interval. If $\kappa > 0$ in $(s_0, s_0 + \varepsilon)$, then γ winds counterclockwise in this interval, and thus $\langle \gamma - \gamma(s_0), n \rangle > 0$ in some interval $(s_0, s_0 + \varepsilon_1)$, because (t, n) is right-handed. From $0 = a + \langle b, \gamma(s_0) \rangle$, it follows that



$$K\kappa = a + \langle b, \gamma \rangle = \langle b, \gamma - \gamma(s_0) \rangle = \langle b, n \rangle \langle \gamma - \gamma(s_0), n \rangle.$$

From $K\kappa > 0$, we see that $\langle b, n \rangle > 0$. The case $\kappa < 0$ in $(s_0, s_0 + \varepsilon)$ proceeds analogously: Then, γ winds clockwise in $(s_0, s_0 + \varepsilon)$, so $\langle \gamma - \gamma(s_0), n \rangle < 0$ in some interval $(s_0, s_0 + \varepsilon_1)$. Because $K\kappa < 0$, we have $\langle b, n \rangle > 0$ again. This shows the claim. □⌋

Because $\langle b, n \rangle > 0$, and K is bounded from below by a positive constant, there exists $A > 0$ such that for all $s \in (s_0, s_0 + \varepsilon_1)$,

$$\frac{\langle \gamma(s) - \gamma(s_0), n \rangle}{\kappa(s)} = \frac{K(s)}{\langle b, n \rangle} \geq A.$$

By Lemma 11, we know that

$$\liminf_{s \to s_0} \frac{\langle \gamma(s) - \gamma(s_0), n \rangle}{\langle \gamma''(s), n \rangle} = \liminf_{s \to s_0} \frac{\langle \gamma(s) - \gamma(s_0), n \rangle}{\kappa(s)} \geq A, \tag{A.1}$$

and that $\langle \gamma'', n \rangle$ does not change sign in some interval $(s_0, s_0 + \varepsilon]$. Let s_ε be a maximizer of $|\langle \gamma'', n \rangle|$ in $(s_0, s_0 + \varepsilon]$. Note that a maximizer exists because $|\langle \gamma'', n \rangle|$ is continuous, and the supremum is not attained at s_0 because $\langle \gamma''(s_0), n \rangle = 0$. Then,

$$|\langle \gamma(s_\varepsilon) - \gamma(s_0), n \rangle| = \int_{s_0}^{s_\varepsilon} \int_{s_0}^{s_1} |\langle \gamma''(s_2), n \rangle| \, \mathrm{d}s_2 \, \mathrm{d}s_1 \leq \varepsilon^2 |\langle \gamma''(s_\varepsilon), n \rangle|,$$

where the equality follows from applying the fundamental theorem of calculus twice, and the inequality from bounding the integration area by ε^2, and the integrand by its maximum. This shows that $\frac{|\langle \gamma(s) - \gamma(s_0), n \rangle|}{|\langle \gamma''(s), n \rangle|}$ becomes arbitrarily small close to s_0, which implies

$$\liminf_{s \to s_0} \frac{\langle \gamma(s) - \gamma(s_0), n \rangle}{\langle \gamma''(s), n \rangle} = \liminf_{s \to s_0} \frac{|\langle \gamma(s) - \gamma(s_0), n \rangle|}{|\langle \gamma''(s), n \rangle|} = 0.$$

This contradicts Eq. A.1, and our indirect assumption is proven false. We conclude that L is not tangent to γ in s_0.

Proof of (2): To show secant-boundedness of κ at $s_0 \in S_0$, write[1]

$$\gamma'(s_0 + h) = \gamma'(s_0) + h\gamma''(s_0) + o(h) = t + o(h),$$

and, by integrating, $\gamma(s_0 + h) = \gamma(s_0) + ht + o(h^2)$. The equilibrium equation gives

$$K(s_0 + h)\kappa(s_0 + h) = a + \langle b, \gamma(s_0) + ht + o(h^2)\rangle = h\langle b, t\rangle + o(h^2).$$

Because of the non-tangency property, we have $\langle b, t\rangle \neq 0$, so the right-hand side is secant-bounded at $h = 0$. On the other hand, K is bounded from below and above by positive constants. Thus, κ is secant-bounded at s_0.

Direction "$\Leftarrow$": We assume that (1) and (2) hold and show the existence of K admissible and $a \in \mathbb{R}$, $b \in \mathbb{R}^2$ that solve $K\kappa = a + \langle b, \gamma\rangle$. Because κ is secant-bounded on S_0, it changes sign at every $s_0 \in S_0$. Because of the non-tangency condition, γ crosses from one side of L to the other at every $s_0 \in S_0$. Thus, the portion of the curve with $\kappa > 0$ lies fully on one side of L, and the portion with $\kappa < 0$ lies fully on the other. Choose a and b such that $L = \{x \in \mathbb{R}^2 : a + \langle b, x\rangle = 0\}$ and such that $a + \langle b, x\rangle > 0$ on the same side of L as $\kappa > 0$.

Then, formally set $K(s) := \frac{a+\langle b,\gamma(s)\rangle}{\kappa(s)}$ for $s \in [0, l]$. Away from S_0, this function is positive and continuous, and thus bounded from above. Furthermore, κ is continuous on $[0, l]$ and thus bounded, which implies that K is also bounded from below by a positive constant, away from S_0.

It remains to show that K is also bounded from above and below by positive constants as $s \to s_0 \in S_0$. By the same argument as above, we have

$$K(s_0 + h) = \frac{\langle b, t\rangle h + g(h)}{\kappa(s_0 + h)}, \quad \text{where} \quad g(h) = o(h^2). \tag{A.2}$$

Both numerator and denominator are secant-bounded at $h = 0$, and the signs of their secant bounds coincide. This is enough to guarantee that K has the desired bounds. To see this, one analyzes K as $h \to 0^+$ and $h \to 0^-$, for the cases where the secant bounds are either all positive or all negative.

For example, take the all-positive case and analyze $h \to 0^+$: There exist $d, D, e, E > 0$ and $\varepsilon > 0$ such that $dh < \langle b, t\rangle h + g(h) < Dh$ and $eh < \kappa(s_0 + h) < Eh$ for all $h \in (0, \varepsilon)$. Then, we can bound $d/E < K(s_0 + h) < D/e$. This shows that K is admissible. □

A.2 Moment Equilibrium Under Gravity

In Section 5.1 of the main article, we give an expression for the energy potential of an elastic curve with constant thickness and spatially-varying width under gravity,

$$W[\alpha] = \int_0^l K\left(\frac{1}{2}\kappa^2 + \langle \gamma, e\rangle\right),$$

[1]We write $o(z(h))$ as a shorthand for some function $g(h)$ such that $\lim_{h\to 0} g(h)/z(h) = 0$. This means that $g(h)$ decays strictly faster than $z(h)$ as $h \to 0$. Therefore, if $f \in C^1$, then $f(x+h) = f(x)+hf'(x)+o(h)$.

where $e \in \mathbb{R}^2$ is a constant dependent on gravity, thickness, and material properties. We are looking for extremals subject to the constraint $\gamma(l) = \gamma_l$, i.e., the endpoint of the curve is fixed to lie at γ_l. According to the method of Lagrange multipliers, these extremals are characterized by α and $\lambda \in \mathbb{R}^2$ for which the variation of

$$L[\alpha, \lambda] = \underbrace{\int_0^l \frac{1}{2} K\kappa^2}_{A} + \underbrace{\int_0^l K\langle \gamma, e\rangle}_{B} + \underbrace{\int_0^l \langle \lambda, \gamma'\rangle}_{C}$$

vanishes, and such that $\gamma(l) = \gamma_l$. The variations of A, B, and C are given by

$$\begin{aligned}
\delta A &= \int_0^l K\kappa\delta\alpha' = -\int_0^l (K\kappa)'\delta\alpha, \\
\delta B &= \int_0^l K(s)\left(\int_0^s \langle R\gamma', e\rangle\delta\alpha\right)\mathrm{d}s, \\
\delta C &= \int_0^l \langle \lambda, R\gamma'\rangle\delta\alpha.
\end{aligned}$$

To derive the moment equilibrium equation at a point $\varsigma \in (0, l)$, we formally set $\delta\alpha = \delta_\varsigma$, the delta distribution centered at ς. Then, the variations of A and C evaluate to

$$\delta A = -(K\kappa)'(\varsigma) \quad \text{and} \quad \delta C = \langle \lambda, R\gamma'(\varsigma)\rangle.$$

The variation of B is given by

$$\begin{aligned}
\delta B &= \int_0^l K(s)\langle R\gamma'(\varsigma), e\rangle\chi_{[0,s]}(\varsigma)\,\mathrm{d}s \\
&= \langle R\gamma'(\varsigma), e\rangle \int_0^l K(s)\chi_{[0,s]}(\varsigma)\,\mathrm{d}s = \langle R\gamma'(\varsigma), e\rangle \int_\varsigma^l K,
\end{aligned}$$

where $\chi_{[0,s]}$ denotes the characteristic function on $[0, s]$. Setting the variation of L to zero, we arrive at the Euler–Lagrange equation

$$0 = -(K\kappa)'(\varsigma) + \langle b, \gamma'(\varsigma)\rangle + \langle R^t e, \gamma'(\varsigma)\rangle \int_\varsigma^l K,$$

where we have substituted $\lambda = Rb$.

The last step in deriving the moment equilibrium equation is to find an antiderivative of the function above. For the first two summands, an antiderivative is given by $-K\kappa + \langle b, \gamma\rangle$. For the last summand, we use the identity

$$\frac{\mathrm{d}}{\mathrm{d}s}\left(f(s)\int_s^l g - \int_s^l fg\right) = f'(s)\int_s^l g$$

to conclude that an antiderivative is given by

$$\langle R^t e, \gamma(\varsigma)\rangle \int_\varsigma^l K - \int_\varsigma^l \langle R^t e, \gamma\rangle K.$$

In its integrated form, the moment equilibrium equation is given by

$$-K(\varsigma)\kappa(\varsigma) + \langle b, \gamma(\varsigma)\rangle + \langle R^t e, \gamma(\varsigma)\rangle \int_\varsigma^l K - \int_\varsigma^l \langle R^t e, \gamma\rangle K + a = 0$$

with the integration constant a.

A.3 Adjoint Equations for Conjugate Points

A.3.1 Summary

We show how to compute $\partial\mathcal{Z}/\partial K|_{(\sigma,K)}$ by applying the adjoint method to the Euler–Lagrange equations and the Jacobi criterion. Let us define the functional $F[K] := \int \delta_\sigma(s)\mathcal{Z}(s,K)\,\mathrm{d}s$ with δ_σ the delta distribution centered at σ, so $\delta F/\delta K = \partial\mathcal{Z}/\partial K|_{(\sigma,K)}$. For easier implementation, δ_σ can also be replaced with a bump function that has small support, is centered at σ, and integrates to one.

The cofactors of the matrix $Z = \begin{pmatrix} \zeta & \eta_1 & \eta_2 \\ M_1 & N_{11} & N_{12} \\ M_2 & N_{21} & N_{22} \end{pmatrix}$ appear in the adjoint equations, and we will denote them by $C_{(\cdot)}$, e.g., C_ζ is the cofactor associated with the top-left entry ζ in Z. The adjoint variables will be denoted by overbars. First, we compute $\bar{M}_i$, $\bar{N}_{ij}$, $\bar{\zeta}$, and $\bar{\eta}_i$ via the following sequence of adjoint equations:

$$\begin{aligned}
\bar{M}_i' &= \delta_\sigma C_{M_i}, & \bar{M}_i(l) &= 0,\\
\bar{N}_{ij}' &= \delta_\sigma C_{N_{ij}}, & \bar{N}_{ij}(l) &= 0,\\
-(K\bar{\zeta}')' - \langle\lambda,\gamma'\rangle\bar{\zeta} &= \textstyle\sum_k \bar{M}_k T_k - \delta_\sigma C_\zeta, & \bar{\zeta}(l) &= 0,\ \bar{\zeta}'(l) = 0,\\
-(K\bar{\eta}_i')' - \langle\lambda,\gamma'\rangle\bar{\eta}_i &= \textstyle\sum_k \bar{N}_{ki} T_k - \delta_\sigma C_{\eta_i}, & \bar{\eta}_i(l) &= 0,\ \bar{\eta}_i'(l) = 0.
\end{aligned}$$

Then, we solve for the extremal of the variational problem

$$\begin{aligned}
&\int_0^l \frac{1}{2}\left(K\bar{\alpha}'^2 - \langle\lambda,\gamma'\rangle\bar{\alpha}^2\right)\\
&\quad + \Big[\bar{\zeta}\zeta\langle R\lambda,\gamma'\rangle + \textstyle\sum_i \bar{\eta}_i(\eta_i\langle R\lambda,\gamma'\rangle + g_i)\\
&\qquad + \textstyle\sum_i \bar{M}_i g_i\zeta + \sum_{ij}\bar{N}_{ij}g_i\eta_j\Big]\bar{\alpha}
\end{aligned}$$

$$\text{s.t.}\quad \begin{matrix}\bar{\alpha}(0) = 0,\\ \bar{\alpha}(l) = 0,\end{matrix}\quad \text{and}\quad \int_0^l T_i\bar{\alpha} = \int_0^l g_i\Big(\bar{\zeta}\zeta + \textstyle\sum_k\bar{\eta}_k\eta_k\Big)\ \text{for}\ i = 1,2$$

to compute $\bar{\alpha}$. Finally, the variational derivative of F reads

$$\delta F[\delta K] = \Big(\bar{\zeta}(0) + \textstyle\sum_i\bar{\eta}_i(0)\Big)\delta K(0) + \int_0^l\Big(\bar{\alpha}'\alpha' + \bar{\zeta}'\zeta' + \textstyle\sum_i\bar{\eta}_i'\eta_i'\Big)\delta K.$$

A.3.2 Derivation

For easier reference, we repeat all primal equations that need to be taken into account when deriving the adjoint.

$$
\begin{aligned}
&-(K\alpha')' + \langle \lambda, R\gamma' \rangle = 0, && \alpha(0) = \alpha_0, && \alpha(l) = \alpha_l,\\
&\quad \text{subject to} \quad \gamma(l) = \gamma_l,\\
&-(K\zeta')' - \langle \lambda, \gamma' \rangle \zeta = 0, && \zeta(0) = 0, && \zeta'(0) = 1,\\
&-(K\eta_i')' - \langle \lambda, \gamma' \rangle \eta_i = \langle R\gamma', e_i \rangle, && \eta_i(0) = 0, && \eta_i'(0) = 1,\\
&\qquad M_i' = \langle R\gamma', e_i \rangle \zeta, && M_i(0) = 0,\\
&\qquad N_{ij}' = \langle R\gamma', e_i \rangle \eta_j, && N_{ij}(0) = 0,\\
&\qquad Z = \begin{pmatrix} \zeta & \eta_1 & \eta_2 \\ M_1 & N_{11} & N_{12} \\ M_2 & N_{21} & N_{22} \end{pmatrix},\\
&\qquad \mathcal{Z} = \det Z,\\
&\qquad F = \int \varphi \mathcal{Z}.
\end{aligned}
$$

In the following, indices always run from 1 to 2, and integrals from 0 to l. We introduce the adjoint variables $\bar{\alpha}$, $\bar{\lambda}$, $\bar{\zeta}$, $\bar{\eta}_i$, $\bar{M}_i$, and $\bar{N}_{ij}$, corresponding to lines 1–6 of the primal equations. Furthermore, denote by $C_{(\cdot)}$ the cofactors of Z, e.g., C_ζ is the cofactor associated with the top-left entry ζ in the matrix.

The variation of F, before appending additional terms, reads

$$
\delta F = \int \varphi \left(C_\zeta \delta\zeta + \textstyle\sum_i C_{\eta_i} \delta\eta_i + \sum_i C_{M_i} \delta M_i + \sum_{ij} C_{N_{ij}} \delta N_{ij} \right).
$$

Next, let us account for the adjoint terms resulting from expressions of the form $-(Ku')'$, where $u \in \{\alpha, \zeta, \eta_i\}$. This results in

$$
\begin{aligned}
\delta \textstyle\int -\bar{u}(Ku')' = {} & \left(-\bar{u}u'\delta K + K\bar{u}'\delta u - K\bar{u}\,\delta u'\right)\big|_0^l \\
& + \textstyle\int \left(\bar{u}'u'\delta K - (K\bar{u}')'\delta u\right).
\end{aligned}
$$

After considering the boundary conditions on α, ζ, and η_i, we get

$$
\begin{aligned}
\delta \textstyle\int -\bar{\alpha}(K\alpha')' = {} & \left(-\bar{\alpha}\alpha'\delta K - K\bar{\alpha}\,\delta\alpha'\right)\big|_0^l \\
& + \textstyle\int \left(\bar{\alpha}'\alpha'\delta K - (K\bar{\alpha}')'\delta\alpha\right),\\
\delta \textstyle\int -\bar{\zeta}(K\zeta')' = {} & -\bar{\zeta}\zeta'\delta K\Big|_0^l + \left(K\bar{\zeta}'\delta\zeta - K\bar{\zeta}\,\delta\zeta'\right)\Big|_l \\
& + \textstyle\int \left(\bar{\zeta}'\zeta'\delta K - (K\bar{\zeta}')'\delta\zeta\right),\\
\delta \textstyle\int -\bar{\eta}_i(K\eta_i')' = {} & -\bar{\eta}_i\eta_i'\delta K\big|_0^l + \left(K\bar{\eta}_i'\delta\eta_i - K\bar{\eta}_i\,\delta\eta_i'\right)\big|_l \\
& + \textstyle\int \left(\bar{\eta}_i'\eta_i'\delta K - (K\bar{\eta}_i')'\delta\eta_i\right).
\end{aligned}
$$

The remaining term in line 1 and the equality constraint in line 2 yield

$$
\begin{aligned}
\delta \textstyle\int \bar{\alpha}\langle \lambda, R\gamma' \rangle &= \textstyle\int \bar{\alpha}\left(\langle \delta\lambda, R\gamma' \rangle - \langle \lambda, \gamma'\delta\alpha \rangle\right)\\
&= \langle \delta\lambda, \textstyle\int R\gamma'\bar{\alpha} \rangle - \int \bar{\alpha}\langle \lambda, \gamma' \rangle \delta\alpha,\\
\delta\langle \bar{\lambda}, \textstyle\int \gamma' - \gamma_l \rangle &= \textstyle\int \langle \bar{\lambda}, R\gamma' \rangle \delta\alpha.
\end{aligned}
$$

The remaining term in line 3 yields

$$\delta \int -\bar{\zeta}\langle \lambda, \gamma' \rangle \zeta = -\langle \delta\lambda, \int \bar{\zeta}\zeta\gamma' \rangle - \int \bar{\zeta}\zeta \langle \lambda, R\gamma' \rangle \delta\alpha - \int \bar{\zeta}\langle \lambda, \gamma' \rangle \delta\zeta,$$

and the ones in line 4,

$$\begin{aligned}\delta \int -\bar{\eta}_i \left(\langle \lambda, \gamma' \rangle \eta_i + \langle R\gamma', e_i \rangle \right) \\ = -\langle \delta\lambda, \int \bar{\eta}_i \eta_i \gamma' \rangle + \int \bar{\eta}_i \langle Re_i - \lambda\eta_i, R\gamma' \rangle \delta\alpha - \int \bar{\eta}_i \langle \lambda, \gamma' \rangle \delta\eta_i.\end{aligned}$$

Similarly, for lines 5 and 6,

$$\begin{aligned}\delta \int \bar{M}_i \left(M_i' - \langle R\gamma', e_i \rangle \zeta \right) \\ = \bar{M}_i \, \delta M_i \Big|_l - \int \bar{M}_i' \delta M_i + \int \bar{M}_i \zeta \langle \gamma', e_i \rangle \delta\alpha - \int \bar{M}_i \langle R\gamma', e_i \rangle \delta\zeta,\end{aligned}$$

$$\begin{aligned}\delta \int \bar{N}_{ij} \left(N_{ij}' - \langle R\gamma', e_i \rangle \eta_j \right) \\ = \bar{N}_{ij} \, \delta N_{ij} \Big|_l - \int \bar{N}_{ij}' \delta N_{ij} + \int \bar{N}_{ij} \eta_j \langle \gamma', e_i \rangle \delta\alpha - \int \bar{N}_{ij} \langle R\gamma', e_i \rangle \delta\eta_j.\end{aligned}$$

Next, we gather the expressions multiplying variations that we do not want to evaluate, i.e., $\delta\alpha$, $\delta\lambda$, $\delta\zeta$, $\delta\eta_i$, δM_i, and δN_{ij}. We will do this process in reverse order, because this corresponds to the natural order in which the adjoint equations need to be solved. For δM_i and δN_{ij}, we have

$$\begin{aligned} \bar{M}_i' &= \varphi C_{M_i}, & \bar{M}_i(l) &= 0, \\ \bar{N}_{ij}' &= \varphi C_{N_{ij}}, & \bar{N}_{ij}(l) &= 0. \end{aligned}$$

For $\delta\zeta$ and $\delta\eta_i$, we have

$$\begin{aligned} -(K\bar{\zeta}')' - \langle \lambda, \gamma' \rangle \bar{\zeta} &= \bar{M}_i \langle R\gamma', e_i \rangle - \varphi C_\zeta, & \bar{\zeta}(l) &= 0, & \bar{\zeta}'(l) &= 0, \\ -(K\bar{\eta}_i')' - \langle \lambda, \gamma' \rangle \bar{\eta}_i &= \bar{N}_{ij} \langle R\gamma', e_i \rangle - \varphi C_{\eta_i}, & \bar{\eta}_i(l) &= 0, & \bar{\eta}_i'(l) &= 0. \end{aligned}$$

Collecting terms multiplying $\delta\alpha$ yields

$$\begin{aligned}(K\bar{\alpha}')' = &-\bar{\alpha}\langle \lambda, \gamma' \rangle + \langle \bar{\lambda}, R\gamma' \rangle \\ &+ (\bar{\zeta}\zeta + \textstyle\sum_i \bar{\eta}_i \eta_i) \langle R\lambda, \gamma' \rangle \\ &+ \textstyle\sum_i (\bar{\eta}_i + \bar{M}_i \zeta + \sum_j \bar{N}_{ij} \eta_j) \langle \gamma', e_i \rangle, \qquad \bar{\alpha}(0) = 0, \ \bar{\alpha}(l) = 0,\end{aligned}$$

and for $\delta\lambda$, the two-component equation

$$\int R\gamma' \bar{\alpha} = \int (\bar{\zeta}\zeta + \textstyle\sum_i \bar{\eta}_i \eta_i) \gamma'.$$

Putting the equations for $\delta\alpha$ and $\delta\lambda$ together, we see that they correspond to the constrained Euler–Lagrange equations of a quadratic variational problem with linear integral constraints in $\bar{\alpha}$, where $\bar{\lambda}$ is used as a Lagrange multiplier:

$$\begin{aligned}&\int_0^l \frac{1}{2} \left(K\bar{\alpha}'^2 - \langle \lambda, \gamma' \rangle \bar{\alpha}^2 \right) \\ &\quad + \Big[(\bar{\zeta}\zeta + \textstyle\sum_i \bar{\eta}_i \eta_i) \langle R\lambda, \gamma' \rangle \\ &\qquad + \textstyle\sum_i (\bar{\eta}_i + \bar{M}_i \zeta + \sum_j \bar{N}_{ij} \eta_j) \langle \gamma', e_i \rangle \Big] \bar{\alpha} \\ \text{s.t.} \quad &\begin{matrix} \bar{\alpha}(0) = 0, \\ \bar{\alpha}(l) = 0, \end{matrix} \quad \text{and} \quad \int_0^l R\gamma' \bar{\alpha} = \int_0^l (\bar{\zeta}\zeta + \textstyle\sum_i \bar{\eta}_i \eta_i) \gamma'.\end{aligned}$$

Finally, the variational derivative of F can be assembled from all terms involving δK. This gives

$$\delta F[K;\delta K] = (\bar{\zeta} + \textstyle\sum_i \bar{\eta}_i)\,\delta K\,\Big|_0 + \int_0^l \big(\bar{\alpha}'\alpha' + \bar{\zeta}'\zeta' + \textstyle\sum_i \bar{\eta}_i'\eta_i'\big)\,\delta K.$$

A.4 Constraint Satisfaction

Algorithm 4.2 calls the routine EnforceConstraints, which acts on the control points of a spline curve. Its objective is to restore the equilibrium property of the curve through collinearity of inflection points, and to enforce boundary conditions and fixed arc length.

We assume a spline parametrization $\gamma(t,q) = \sum_i B_i(t)\,q_i$ with B_i piecewise polynomial and control points $q_1,\dots,q_m \in \mathbb{R}^2$. For a total number of N constraints, we denote the constraint manifold as $G(q) = 0$ with $G : \mathbb{R}^{2m} \to \mathbb{R}^N$, and its Jacobian as $J_G : \mathbb{R}^{2m} \to \mathbb{R}^{N\times 2m}$. Rows of J_G corresponding to linear constraints may be precomputed at the beginning, and the remaining rows are updated every time the routine is called. First, we discuss all constraint types, and then the Newton-type iteration by which they are enforced.

Collinearity of Inflections We over-constrain inflection points by keeping the inflection line L constant during optimization. For each inflection, this produces one equality constraint of the form $G_{\mathsf{infl}}(q) := a + \langle \gamma(t_0(q),q), b\rangle = 0$, where t_0 depends on q implicitly via $\det(\gamma'(t_0,q),\gamma''(t_0,q)) = 0$. Derivatives can be evaluated as

$$\begin{aligned}
\frac{\partial t_0}{\partial q_{i,j}} &= -\frac{1}{\det(\gamma',\gamma''')}\left[\det\left(\partial\gamma'/\partial q_{i,j},\gamma''\right) + \det\left(\gamma',\partial\gamma''/\partial q_{i,j}\right)\right],\\
\frac{\partial G_{\mathsf{infl}}}{\partial q_{i,j}} &= \left\langle \frac{\partial\gamma}{\partial q_{i,j}} + \gamma'\frac{\partial t_0}{\partial q_{i,j}}, b\right\rangle,
\end{aligned}$$

where $q_{i,j}$ represents the j-th coordinate of q_i. Note that regularity of the constraint $G_{\mathsf{infl}}(q) = 0$ is guaranteed by the upper bound on $R[K]$ that is introduced in Algorithm 4.2.

Boundary Conditions Boundary points are fixed by constraints $\gamma(0) = \gamma_0$ and $\gamma(l) = \gamma_l$, and tangents by constraining $\langle\gamma'(0), n_0\rangle = 0 = \langle\gamma'(l), n_l\rangle$, where n_0 and n_l are the initial normals to the curve at its endpoints. These constraints are linear in q, so the derivatives $\partial\gamma/\partial q_{i,j}$ and $\partial\gamma'/\partial q_{i,j}$ can be precomputed at the beginning.

Fixed Arc Length The arc length of a curve is discretized as $l(q) = \sum_{i=0}^{n-1} \|\gamma(t_{i+1},q) - \gamma(t_i,q)\|$, and fixed with a constraint $G_{\mathsf{arc}}(q) := l(q) - l_0 = 0$. Here $t_0,\dots,t_n$ is a sampling of the parameter domain, and l_0 is the initial arc length.

Enforcing Constraints The input to EnforceConstraints is a set of control points $q^0 \in \mathbb{R}^{2m}$ that might violate the constraints, and the goal is to find q^n such that $G(q^n) = 0$, and q^n close to q^0. We achieve this using an underdetermined Newton iteration $q^{i+1} = q^i + \Delta q$, where Δq is the least-norm solution to $J_G\,\Delta q = -G$. This iteration does not converge to the orthogonal projection of q^0 onto $\{q \in \mathbb{R}^{2m} : G(q) = 0\}$, but it is a good approximation that can be computed robustly.

A.5 Nonlinear Materials

Our design system can be extended to account for certain material nonlinearities. We have not explored these models in our physical results, except for the spiral example shown in Fig. 4.7, which uses plasticity to account for the curvature of the innermost winding.

Nonlinear Elasticity The left-hand side of Eq. 4.3 represents the internal moment $M(s) = K(s)\kappa(s)$ integrated over a cross section of an elastic strip. We can further decompose $K(s) = P\,w(s)$, where $w(s)$ is the width of the strip, and P represents a linear material law.

A wide class of nonlinear material laws take the similar form

$$M(s) = P(\kappa(s))\,w(s)\,\kappa(s),$$

for $P : \mathbb{R} \to \mathbb{R}_{>0}$ an even function, and thus $P'(0) = 0$. This class includes for example strain-hardening behaviors ($P''(0) > 0$), and strain-softening behaviors ($P''(0) < 0$). Perhaps surprisingly, the theory developed in Section 4.3 applies to these laws as well, as long as the moment-per-unit-width function $m(\kappa) := P(\kappa)\kappa$ is injective in the relevant curvature range. If this is the case, we can solve for $a \in \mathbb{R}$ and $b \in \mathbb{R}^2$ using the same linear program, and then compute $w(s)$ from the equation $w(s)\,m(\kappa(s)) = a + \langle b, \gamma(s)\rangle$.

Plasticity If an elastic strip is bent beyond the elastic limit of the base material, some of the deformation will become permanent, and the strip does not return to its original flat state after removing external forces. This effect was noticeable in our experiments with bending cardboard (200 gsm) for curvature radii below 2 cm.

The simplest plasticity model is that of an *ideally plastic* material, which assumes that the linear stress-strain law is replaced by a constant law at the elastic limit. Applied to elastic strips, this model postulates the existence of a curvature limit κ_{lim}, such that all curvature beyond this point becomes plastic. We implemented this model in our design system by replacing $\kappa(s_i)$ in Eq. 4.4 by an *effective curvature* $\kappa_{\mathsf{eff}}(s_i) := \min\{\kappa(s_i), \kappa_{\mathsf{lim}}\}$. This implementation does not account for path-dependent deformation during the bending process, or the formation of plastic hinges. As such, it is only an approximation of ideally plastic behavior.

APPENDIX B

Kirchhoff Rods

B.1 Proof of Proposition 2

We show that zero is the tight lower bound of the torsional rigidity for any given bending rigidity. In other words, given $I \in S^2_{++}$ and $\varepsilon > 0$, we can find a bounded (and simply-connected) domain $\mathcal{D} \subset \mathbb{R}^2$ with bending rigidity equal to I and torsional rigidity at most ε.

Choose $I \in S^2_{++}$ and $r > 0$, and construct a domain $\Omega \subset \mathbb{R}^2$ as follows: Starting from an elliptical disk with bending rigidity I, add linear cuts from the boundary to the interior in such a way that the domain remains simply connected and that the incircle radius (the supremum of the radii of all circles contained in Ω) falls below r, as illustrated in Fig. 5.6 (right). Let $\chi \in H^1_0(\Omega)$ be the solution to $\Delta\chi = -1$ in Ω and $\chi = 0$ on $\partial\Omega$, where $H^1_0(\Omega)$ denotes the Sobolev space of weakly differentiable functions in $L^2(\Omega)$ supported in Ω.

Partition the axis-aligned bounding rectangle of Ω into a rectilinear grid such that every cell has side lengths greater than $2r$ and at most $3r$ (which is always possible for small enough r). This guarantees that every cell intersects Ω^c in a set containing a line segment of positive length, on which $\chi = 0$. By the Poincaré–Friedrichs inequality [Bra07, II.1.5-6], we have

$$\|\chi\|_{L^2(C)} \leq 3r\,\|\nabla\chi\|_{L^2(C)}$$

for every cell C of the partition, and by summing over all cells,

$$\|\chi\|_{L^2(\Omega)} \leq 3r\,\|\nabla\chi\|_{L^2(\Omega)}.$$

By Green's first identity, we have

$$\|\nabla\chi\|^2_{L^2(\Omega)} = \int_{\partial\Omega} \chi\,\partial_n\chi - \int_\Omega \chi\,\Delta\chi = \int_\Omega \chi \leq \|\chi\|_{L^1(\Omega)},$$

and by the Cauchy–Schwarz inequality,

$$\|1\cdot\chi\|_{L^1(\Omega)} \leq \|1\|_{L^2(\Omega)}\|\chi\|_{L^2(\Omega)} = \sqrt{\mu(\Omega)}\|\chi\|_{L^2(\Omega)},$$

where μ denotes the Lebesgue measure. Altogether, this gives

$$\|\nabla\chi\|^2_{L^2(\Omega)} \leq \|\chi\|_{L^1(\Omega)} \leq \sqrt{\mu(\Omega)}\|\chi\|_{L^2(\Omega)} \leq 3r\sqrt{\mu(\Omega)}\|\nabla\chi\|_{L^2(\Omega)}.$$

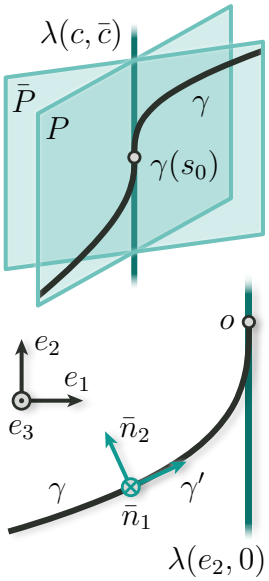
Cancelling $\|\nabla\chi\|_{L^2(\Omega)}$ and squaring gives

$$J = 4\|\nabla\chi\|^2_{L^2(\Omega)} \leq 36r^2\mu(\Omega).$$

Choosing r such that $36r^2\mu(\Omega) \leq \varepsilon$ gives the statement. □

B.2 Proof of Proposition 7 (cont.)

It remains to show that the case $\langle c, \bar{c} \rangle = 0$ with $c \neq 0$ reduces to Theorem 6. We know by assumption that all tangent lines of γ intersect the Euclidean line $\lambda(c, \bar{c})$, as shown in Fig. 5.8 (left). This implies that every connected component of $\gamma((0, \ell)) \setminus \lambda(c, \bar{c})$ is contained in a plane P with $\lambda(c, \bar{c}) \subset P$. The points $\gamma(s_0) \in \lambda(c, \bar{c})$ are exactly the inflection points of γ, so there are finitely many connected components and planes.



Consider an interval $(s_0 - \varepsilon, s_0 + \varepsilon)$ small enough such that $\gamma((s_0 - \varepsilon, s_0))$ is contained in a plane P, and $\gamma((s_0, s_0 + \varepsilon))$ in a plane $\bar{P}$. Assume for the sake of contradiction that $P \neq \bar{P}$, so $P \cap \bar{P} = \lambda(c, \bar{c})$. Then, the tangent line of γ at s_0 must coincide with $\lambda(c, \bar{c})$.

Assume wlog that $\gamma(s_0)$ coincides with the origin, $(c, \bar{c}) = (e_2, 0)$, and P is normal to e_3. Next, consider the parallel frame $\bar{F} = (\bar{n}_1, \bar{n}_2, \gamma')$ such that $\bar{n}_1 \equiv -e_3$ on the interval $(s_0 - \varepsilon, s_0)$. This frame is related to F by a constant rotation $Q \in SO(2)$, and satisfies the equilibrium equation with $\bar{I} = Q^t I Q$:

$$E \bar{F}_n \bar{I} \bar{k}_n = E(F_n Q)(Q^t I Q)(Q^t k_n) = E F_n I k_n = c \times \gamma + \bar{c},$$

according to Eq. 5.1. Noting that $\bar{\kappa}_2 \equiv 0$, $\langle \bar{n}_2, e_1 \rangle = -\langle \gamma', e_2 \rangle$, and $\langle \bar{n}_2, e_2 \rangle = \langle \gamma, e_1 \rangle$ on $(s_0 - \varepsilon, s_0)$, we can write out the equilibrium equation in coordinates, which yields

$$\bar{I}_{xy} \bar{\kappa}_1 \langle \gamma', e_1 \rangle = 0 = \bar{I}_{xy} \bar{\kappa}_1 \langle \gamma', e_2 \rangle, \quad E \bar{I}_{xx} \bar{\kappa}_1 = \langle \gamma, e_1 \rangle.$$

The two equations on the left imply $\bar{I}_{xy} \bar{\kappa}_1 \equiv 0$ and thus $\bar{I}_{xy} \equiv 0$ because $\bar{\kappa}_1(s) = 0$ only at $s = s_0$. The equation on the right is exactly the equilibrium equation from Theorem 6, with $\lambda(e_2, 0)$ equivalent to L. That $\lambda(e_2, 0)$ is tangent to γ contradicts Theorem 6(1), and shows $P = \bar{P}$. By repeating the argument, we see that every connected component of $\gamma((0, \ell)) \setminus \lambda(c, \bar{c})$ is contained in the same plane. □

B.3 Helical Symmetry of Constant-Curvature Parallel Equilibrium Curves

We give a sketch of the proof that solutions to

$$\gamma''(s) = B(\gamma(s)) \times \gamma'(s), \quad \text{with} \quad B(x) = \kappa \frac{c \times x + \bar{c}}{\| c \times x + \bar{c} \|},$$

initial conditions given in Eq. 5.15, and constant $\kappa > 0$ have a discrete helical symmetry with axis e_3, i.e., there exist $h, \zeta \in \mathbb{R}$ and $\sigma > 0$, such that, for all $s \in \mathbb{R}$,

$$\gamma(s + \sigma) = \begin{pmatrix} \cos\zeta & -\sin\zeta & 0 \\ \sin\zeta & \cos\zeta & 0 \\ 0 & 0 & 1 \end{pmatrix} \gamma(s) + \begin{pmatrix} 0 \\ 0 \\ h \end{pmatrix}.$$

To show this, we note that B is divergence-free, so the equation $\gamma'' = B \times \gamma'$ describes the trajectory of a charged particle in a magnetic field. In cylindrical coordinates with radius ϱ, azimuth θ, and height z, we define the vector potential

$$A(\varrho, \theta) = \frac{\kappa p}{\varrho} \left(\sqrt{\varrho^2 + p^2} - 1 \right) e_\theta(\theta) + \kappa \left(\sqrt{\varrho^2 + p^2} - 1 \right) e_3,$$

such that $\operatorname{div} A = 0$ and $\operatorname{curl} A = B$, where B is given by

$$B(\varrho, \theta) = \frac{\kappa}{\sqrt{\varrho^2 + p^2}} \left(\varrho e_\theta(\theta) + p e_3\right)$$

in cylindrical coordinates. The Lagrangian for a charged particle in a magnetic field is given by $\frac{1}{2}\langle \gamma', \gamma' \rangle + \langle \gamma', A \circ \gamma \rangle$, which we can use to extract invariants of our differential equation by using Noether's theorem. Symmetry under time translation gives the arc-length condition $\varrho'^2 + (\varrho\theta')^2 + z'^2 = 1$, and symmetry under translation along e_3 and rotation around e_3 gives two new invariants

$$I_z = z' + \kappa\sqrt{\varrho^2 + p^2}, \quad I_\theta = \varrho^2\theta' - \kappa p\sqrt{\varrho^2 + p^2},$$

which show that z' and θ' only depend on ϱ (but not on θ and z). We compute $0 = \langle \gamma', c \times \gamma + \bar{c} \rangle = \varrho^2\theta' + pz' = I_\theta + pI_z$, showing that $I_\theta = -pI_z$. Next, we substitute θ' and z' in the arc-length condition for the invariants, which gives (for $\varrho \neq 0$),

$$\varrho'^2 = 1 - \frac{\varrho^2 + p^2}{\varrho^2}\left(I_z - \kappa\sqrt{\varrho^2 + p^2}\right)^2, \tag{B.1}$$

showing that ϱ' only depends on ϱ, up to sign. To determine zeros of ϱ', substitute $\bar{\varrho} = \sqrt{\varrho^2 + p^2} \geq |p|$, which gives

$$\bar{\varrho}^2(1 - (I_z - \kappa\bar{\varrho})^2) = p^2.$$

On $\bar{\varrho} > |p|$, this equation has either two distinct real solutions, or one real double solution (which corresponds to the special case of a single-helical solution). In the former case, we have $\varrho'(\varrho) = 0$ exactly for some $\varrho = R_1$ and $\varrho = R_2$, with $0 < R_1 < R_2$. Then, ϱ consists of alternating, mirror-symmetric segments, on which ϱ monotonically increases from R_1 to R_2, and then monotonically decreases from R_2 to R_1. To show that the sign of ϱ' actually changes at R_1 and R_2, one verifies $\varrho'' \neq 0$ at these points. Furthermore, θ' and z' only depend on ϱ, so every pair of alternating segments will give a copy of the same curve segment, which is translated along and rotated around e_3 with respect to the previous one. This shows the discrete helical symmetry of γ.

B.4 Proof of Proposition 10

$(1) \Rightarrow (2)$: By (1), we assume that there exists $(I, J) : (0, \ell) \to \mathcal{K}$ satisfying the equilibrium equation $EF_nIk_n + \mu J\tau\gamma' = c \times \gamma + \bar{c}$, such that J and the eigenvalues of I are bounded from below and above by positive constants. Furthermore, $\kappa = \|k_n\|$ is also bounded from below and above by positive constants, because it is positive and continuous on $[0, \ell]$.

Taking the inner product between the equilibrium equation and γ' yields $\mu J\tau = \langle \gamma', c \times \gamma + \bar{c} \rangle$, which shows that $\langle \gamma'/\tau, c \times \gamma + \bar{c} \rangle$ has the required properties by the coercivity and boundedness of J.

To show the second inequality, take the inner product between the equilibrium equation and $\omega_n/(E\kappa^2)$, which yields

$$\lambda_1 \leq \left\langle \frac{k_n}{\kappa}, I\frac{k_n}{\kappa} \right\rangle = \left\langle \frac{\omega_n}{E\kappa^2}, c \times \gamma + \bar{c} \right\rangle \leq \lambda_2,$$

with $\lambda_1 \leq \lambda_2$ the eigenvalues of I. Furthermore, from

$$J \leq 4\psi(I) \leq 4\lambda_1 \frac{\lambda_2}{\lambda_1 + \lambda_2}$$

and the coercivity and boundedness of I, we find that there exists $\varepsilon > 0$ such that $J + \varepsilon \leq 4\lambda_1$. Combining these inequalities with the expression for $\mu J \tau$ from above, we get

$$\varepsilon = (J + \varepsilon) - J \leq \left\langle \frac{4\omega_n}{E\kappa^2} - \frac{\gamma'}{\mu\tau}, c \times \gamma + \bar{c} \right\rangle \leq 4 \sup \lambda_2,$$

which shows the required bounds.

$(2) \Rightarrow (1)$: Define $J = \langle \frac{\gamma'}{\mu\tau}, c \times \gamma + \bar{c} \rangle$ to satisfy the tangential component of the equilibrium equation. From the limit properties of $\langle \gamma'/\tau, c \times \gamma + \bar{c} \rangle$ at zeros of τ, we see that J is coercive and bounded, and we can extend it with arbitrary positive values at these zeros.

Satisfying the normal component equation $EIk_n = F_n^t(c \times \gamma + \bar{c})$ with an appropriate choice of I is equivalent to satisfying $E\tilde{I}Qk_n = QF_n^t(c \times \gamma + \bar{c})$ for some $Q \in SO(2)$ with an appropriate choice of $\tilde{I}$, because we can transform $\tilde{I} = QIQ^t$. Furthermore, I inherits coerciveness and boundedness from $\tilde{I}$, so it suffices to show these properties for the latter.

We choose $Q \in SO(2)$ such that $Qk_n = \kappa e_1$, so it suffices to find $\tilde{I}$ such that $E\kappa\tilde{I}e_1 = QF_n^t(c \times \gamma + \bar{c})$. Choosing $\tilde{I}_{xx} = \frac{1}{E\kappa^2}\langle \omega_n, c \times \gamma + \bar{c} \rangle$ satisfies the first component of this equation, and we can uniquely determine $\tilde{I}_{xy}$ from the second. Next, we pick $\tilde{I}_{yy} = \frac{J\tilde{I}_{xx} + 4\tilde{I}_{xy}^2}{4\tilde{I}_{xx} - J}$ in order to satisfy $J = 4\psi(\tilde{I})$, which is checked by direct computation.

To verify that $\tilde{I}$ is coercive and bounded, it suffices to show that $\operatorname{tr} \tilde{I}$ and $\det \tilde{I} = \frac{J(\tilde{I}_{xx}^2 + \tilde{I}_{xy}^2)}{4\tilde{I}_{xx} - J}$ are bounded from below and above by positive constants. This can be seen from the formula $2\lambda_{1,2} = \operatorname{tr} \tilde{I} \pm \sqrt{\operatorname{tr}^2 \tilde{I} - 4 \det \tilde{I}}$. For $\tilde{I}_{xx}$, boundedness is clear by continuity and from $4\tilde{I}_{xx} = \langle v_1/\mu + v_2, c \times \gamma + \bar{c} \rangle > 0$. For $\tilde{I}_{yy}$ and $\det \tilde{I}$, boundedness of the numerator is clear from the boundedness of $\tilde{I}_{xx}$ and J, and boundedness of the denominator can be seen from

$$4\tilde{I}_{xx} - J = \left\langle \frac{4\omega_n}{E\kappa^2} - \frac{\gamma'}{\mu\tau}, c \times \gamma + \bar{c} \right\rangle,$$

and the limit properties of the expression on the right-hand side, which follow from (2). We conclude that λ_1 and λ_2 are also bounded from below and above by positive constants, which gives the required coercivity and boundedness of I. □

B.5 Cheat Sheet

Some of the symbols refer to quantities that vary across the length of a beam with arc-length parameter $s \in (0, \ell)$. In the paper, we will often omit the parameter s for brevity, whenever we make an argument that is true at every parameter location. Sometimes, we will also write, e.g., $I \in S_{++}^2$ instead of $I : (0, \ell) \to S_{++}^2$, when it is clear from context that a choice $I(s) \in S_{++}^2$ is made for every $s \in (0, \ell)$.

Sym.	Type	Description
$(\cdot)'$	$(\cdot)' : C^d \to C^{d-1}$	First derivative with respect to arc-length parameter s
$[\cdot]_\times$	$[\cdot]_\times : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$	Transforms a vector $v \in \mathbb{R}^3$ into its "cross product matrix", the skew-symmetric matrix $[v]_\times$ such that $[v]_\times x = v \times x$ for all $x \in \mathbb{R}^3$
a	$a \in \mathbb{R}_{>0}$	Radius of an ellipse, associated with the first semi-axis $(\cos\varphi, \sin\varphi)^t$

b	$b \in \mathbb{R}_{>0}$	Radius of an ellipse, associated with the second semi-axis $(-\sin\varphi, \cos\varphi)^t$
β	$\beta : (0, \ell) \to \mathbb{R}$	Rotation of the normal plane, relating two frames F and F_β adapted to the same curve γ via $F_{\beta,n} = F_n Q_\beta$, with $Q_\beta = \left(\begin{smallmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{smallmatrix}\right)$
$(c, \bar{c})$	$c, \bar{c} \in \mathbb{R}^3$	Homogeneous coordinates of the linear complex $\mathcal{C}$
$\mathcal{C}$	$\mathcal{C} \subset \Lambda_{\mathsf{kl}}$	Set of all lines in $\mathbb{R}^3$ whose Plücker coordinates $(l, \bar{l})$ satisfy $\langle l, \bar{c}\rangle + \langle \bar{l}, c\rangle = 0$
$\mathcal{D}$	$\mathcal{D}(s) \subset \mathbb{R}^2$	Cross section of the Kirchhoff rod at a particular $s \in (0, \ell)$; often assumed to be elliptical
E	$E \in \mathbb{R}_{>0}$	Young's modulus of the base material
e_i	$e_i \in \mathbb{R}^3$	Standard basis vectors $e_1 = (1,0,0)^t$, $e_2 = (0,1,0)^t$, and $e_3 = (0,0,1)^t$
F	$F : (0, \ell) \to SO(3)$	Moving frame adapted to γ; encodes the twist of the Kirchhoff rod deformation; the columns of F are given by $F(s) = (n_1(s), n_2(s), \gamma'(s))$
F_n	$F_n : (0, \ell) \to \mathbb{R}^{3\times 2}$	The matrix of material normals of F, so $F_n = FS = (n_1, n_2)$
f_i	$f_i \in \mathbb{R}^3, i = 1, \dots, n$	Concentrated point load f_i is applied to the centerline of a rod at $\gamma(s_i)$
γ	$\gamma : (0, \ell) \to \mathbb{R}^3$	Arc-length parametrized curve that gives the centerline of a deformed Kirchhoff rod; assumed at least twice continuously differentiable
I	$I : (0, \ell) \to S^2_{++}$	Area moment of inertia tensor of the cross section of the Kirchhoff rod, at a particular $s \in (0, \ell)$, given by $I(s) = \int_{\mathcal{D}(s)} \left(\begin{smallmatrix} y^2 & -xy \\ -xy & x^2 \end{smallmatrix}\right) \mathrm{d}(x, y)$
J	$J : (0, \ell) \to \mathbb{R}_{>0}$	Torsional rigidity of the cross section of the Kirchhoff rod, at a particular $s \in (0, \ell)$; computed as $J(s) = 4\int_{D(s)} \|\nabla\chi\|^2$, where χ is the solution to $\Delta\chi = -1$ in $\mathcal{D}(s)$, and $\chi = 0$ on $\partial\mathcal{D}(s)$
K	$K : (0, \ell) \to \mathbb{R}^{3\times 3}$	Stiffness matrix of the Kirchhoff rod, at a particular $s \in (0, \ell)$; the upper-left two-by-two block is given by EI, and the lower-right entry by μJ; we often use K and the pair (I, J) interchangeably, because E and μ are assumed fixed
k	$k : (0, \ell) \to \mathbb{R}^3$	Curvature vector of the framed curve (γ, F), with components $k = (\kappa_1, \kappa_2, \tau)$; related to F and ω via $\omega = Fk$ and $[k]_\times = F^t F'$
k_n	$k_n : (0, \ell) \to \mathbb{R}^2$	Vector of material curvatures of F, so $k_n = S^t k = (\kappa_1, \kappa_2)^t$
$\mathcal{K}$	$\mathcal{K} \subset S^2_{++} \times \mathbb{R}$	Set of admissible stiffnesses (I, J) that satisfy $0 < J \leq 4\psi(I)$; by abuse of notation, we write $K \in \mathcal{K}$ and $(I, J) \in \mathcal{K}$ interchangeably
$\mathcal{K}^*$	$\mathcal{K}^* \subset \mathcal{K}$	Set of stiffnesses induced by elliptical cross sections, i.e., $J = 4\psi(I)$
κ_i	$\kappa_i : (0, \ell) \to \mathbb{R}$	Material curvatures κ_1 and κ_2 of F; measure bending of the Kirchhoff rod around the material normals n_1 and n_2, respectively
κ	$\kappa : (0, \ell) \to \mathbb{R}_{\geq 0}$	Total (Frenet) curvature of γ, given by $\kappa = \|\gamma''\|$; for any frame F adapted to γ, it holds that $\kappa = \sqrt{\kappa_1^2 + \kappa_2^2}$
ℓ	$\ell \in \mathbb{R}_{>0}$	Length of the Kirchhoff rod

λ	$\lambda(l,\bar{l}) \subset \mathbb{R}^3$	Map from the Plücker coordinates $l, \bar{l} \in \mathbb{R}^3$ with $\langle l, \bar{l} \rangle = 0$ to the line in $\mathbb{R}^3$ incident to the point $\frac{l \times \bar{l}}{\langle l, l \rangle}$ and with direction l
Λ_{kl}	$\Lambda_{\mathsf{kl}} \subset \mathbb{P}^5$	Klein quadric, the set of all points with homogeneous coordinates $(l, \bar{l}) \in \mathbb{R}^6$ satisfying $\langle l, \bar{l} \rangle = 0$; we interpret these points as Plücker coordinates of a line in $\mathbb{R}^3$ with direction v and incident to a point x, such that $(l, \bar{l}) = (v, x \times v)$
M	$M : (0, \ell) \to \mathbb{R}^3$	Accumulated moment on a deformed rod, given by $M = \int_0^s \gamma \times q$
μ	$\mu \in \mathbb{R}_{>0}$	Shear modulus of the base material
n_i	$n_i : (0, \ell) \to \mathbb{R}^3$	Material normals of the moving frame F, so $n_i = Fe_i$ for $i = 1, 2$
ν	$\nu \in (-1, 1/2)$	Poisson's ratio of the base material
ω	$\omega : (0, \ell) \to \mathbb{R}^3$	Darboux vector of the moving frame F; related to F and k via $\omega = Fk$ and $F' = [\omega]_\times F$
p	$p : (0, \ell) \to \mathbb{R}^3$	Line load applied to the centerline of a rod, where $p(s)$ gives the load density at $\gamma(s)$
φ	$\varphi \in \mathbb{R}$	Orientation of ellipse with respect to reference frame; first and second semi-axes are given by $(\cos\varphi, \sin\varphi)^t$ and $(-\sin\varphi, \cos\varphi)^t$ respectively
ψ	$\psi : S^2_{++} \to \mathbb{R}_{>0}$	The determinant-over-trace function $\psi(X) := \frac{\det X}{\operatorname{tr} X}$
Q	$Q : (0, \ell) \to \mathbb{R}^3$	Accumulated load on a rod, given by $Q(s) = \int_0^s q$
q	$q \in \mathscr{D}'((0, \ell); \mathbb{R}^3)$	Load distribution applied to the centerline of a rod, consisting of a line load p and point loads f_i
S	$S \in \mathbb{R}^{3\times 2}$	Selection matrix $S = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$ that extracts the first two columns of a three-column matrix by multiplication from the right, i.e., $(x_1, x_2, x_3)S = (x_1, x_2)$
S^2_{++}	$S^2_{++} \subset \mathbb{R}^{2\times 2}$	Set of all symmetric positive-definite 2-by-2 matrices
$SO(3)$	$SO(3) \subset \mathbb{R}^{3\times 3}$	Set of all rotations of $\mathbb{R}^3$ about the origin
s	$s \in (0, \ell)$	Arc-length parameter of γ
s_i	$s_i \in (0, \ell), i = 1, \ldots, n$	Concentrated point load f_i is applied to the centerline of a rod at $\gamma(s_i)$
τ	$\tau : (0, \ell) \to \mathbb{R}$	Twist of the moving frame F; measures rotation per arc-length unit around γ'