Universidad
Rey Juan Carlos

Tesis Doctoral

Computational Design of Flexible Structures

Autor: **Jesús Pérez Rodríguez**

Director: **Miguel Ángel Otaduy**

Programa de Doctorado en Tecnologías de la Información y las Comunicaciones

Escuela Internacional de Doctorado

2018

Abstract

Computational fabrication technologies have revolutionized manufacturing by offering unprecedented control over the shape and material of the fabricated objects at accessible costs. These technologies allow users to design and create objects with arbitrary properties of motion, appearance or deformation. This rich environment spurs the creativity of designers and produces an increasing demand for computer-aided design tools that alleviate design complexity even for non-expert users.

Motivated by this fact, in this thesis, we address the computational design and automatic fabrication of flexible structures, assemblies of interrelated elements that exhibit elastic behavior. We build upon mechanical simulation and numerical optimization to create innovative computational tools that model the attributes of the fabricated objects, predict their static deformation behavior, and automatically infer design attributes from user-specified goals.

With this purpose, we propose a novel mechanical model for the efficient simulation of flexible rod meshes that avoid using numerical constraints. Then, we devise compact and expressive parameterizations of flexible structures, that naturally produce coherent designs. Our tools implement inverse design functionalities based on a sensitivity-based optimization algorithm, which we further extend to deal with local minimum solutions and highly constrained problems. Additionally, we propose interaction approaches that guide the user through the design process. Finally, we validate all these contributions by developing computer-aided design solutions that facilitate the creation of *flexible rod meshes* and *Kirchhoff-Plateau surfaces.*

In the first part of this work, we overview the relevant foundations of mechanical simulation, analyze the optimization problem that arises from inverse elastic design and discuss alternative solutions. Then, in the second part, we propose a computational method for the design of flexible rod meshes that automatically computes a fabricable design from user-defined deformation examples. Finally, in the last part, we study the design and fabrication of Kirchhoff-Plateau surfaces and present a tool for interactively exploring the space of fabricable solutions.

Acknowledgements

I should thank many people for their care and support during the development of this thesis, but it is late night and, as usual, just a few hours ahead of the deadline. Instead, I will thank a few chosen ones, without whom it wouldn't have been possible to get to this point sane. To all those not in the list, I thank you too!

Miguel, for teaching me to be a better researcher. For his hard work and absolute devotion to his students. For being an advisor to look up to, and for his inspiration during hard times. Apparently, there is no problem that cannot be solved.

The members of the MSLab and GMRV, for the laughs during work hours and discussions at lunch time. For kindly teaching me whatever they were learning, and sharing with me the burden and joy of research. I would like to thank in particular to Rosell, Sergio, Vane, Lobo, Lillo, Rosa, Hector, Valentín, Inés, Alex, Álvaro and their loved ones, for filling all these years with incredibly good memories.

The collaborators that I have had, for inspiring me to achieve much more than I thought I could. Thanks for sharing with me your deep knowledge, and for working hard to make our projects successful. My special thanks to Bernhard, for his continued support, trust and insightful teachings.

My supervisors and fellows during the internships at Disney Zurich, for letting me be part of their amazing group. It was truly an honor to learn from them. Special thanks to Gaurav, for his companionship and humor in a strange place.

My hometown friends, for being still there and not forgetting about me. My parents, Carmen and Francisco, for their unconditional love, care and education, for which I will never be thankful enough. And my sister Teresa, who never ceases to amaze me for her wisdom, her strength, and her contagious energy.

And finally Valeria. Thank you for your infinite support, love and patience. For sharing the laughs and the drama. For helping me to be brave, to stay healthy, to be patient, to be proud, and to be happy. Hers is one half of this thesis, and she deserves all forms of acknowledgements.

And with that, I think it is time to get back to work.

Contents

Chapter 1

Introduction

This thesis deals with the computational design of flexible structures. We build upon mechanical simulation and numerical optimization to develop computer-aided design tools that facilitate the creation of *flexible rod meshes* and *Kirchhoff-Plateau surfaces*. For this purpose, we define compact parameterizations that determine the geometry and material of the structures; propose novel mechanical models that allow us to accurately and efficiently predict their deformed shape in static equilibrium; and devise optimization methods for the automatic inference of design attributes from user-defined functional goals. This chapter provides a brief introduction on these topics, focusing on the challenges that have been faced during the development of the thesis, and the contributions that resulted from its completion.

1.1 Computational design

Computational fabrication technologies like 3D printing offer unprecedented control over the shape and material *attributes* of the fabricated objects, with generally little additional cost dependent on object complexity. This enables an extremely fast transition from virtual prototypes to physical realizations. For these reasons, these technologies have already reached an enormous impact on industrial engineering, where rapid prototyping of all kinds of products, from furniture to vehicle parts, is already leading to cost reductions and efficiency gains. In addition, computational fabrication has also "democratized" production tools and brought them closer to designers, due to their relatively low purchase and material costs. Having full control over the geometry and material complexity enables the production of objects with arbitrary *properties* of motion, appearance or deformation, among others. Designers are ultimately responsible for creating the "design" or pattern for the construction of an object considering aesthetic, functional or economic goals.

However, humans are limited in their ability to handle design complexity. For instance, they may lack the understanding of the underlying process that drives the deformation of an object; or it may be too costly to analytically predict the effect of a texture on the reflectance of the material. In general, regardless of the level of expertise of the design team, a typical product goes through an iterative process of analysis, design, prototyping and evaluation that is extremely time consuming. With computational fabrication technologies being now accessible to the public and massively adopted in industry, there is an increasing demand of computational tools that alleviate the burden of design complexity, coming from potential users with a broad range of expertise levels.

Over the last decade, a large part of the computer graphics community has engaged in responding to this demand, taking advantage of the community's scientific heritage. The pursuit of realistic real-world representations has led to data structures for geometry and attribute representation; algorithms for image analysis, synthesis and simulation; and human-computer interaction techniques. All this interdisciplinary knowledge has been combined into computer-aided design (CAD) solutions with the purpose of facilitating the design of fabricable objects.

Modern computational design solutions offer four major functionalities (Fig. 1.1):

1. *Modeling* design attributes. This implies the definition of a *design space*, i.e., finding a computational representation of the possibly complex geometry and material attributes that determine the fabricated object. In this sense, there is a preference for exhaustive but compact models that naturally produce "good" designs yet facilitate user control.

2. *Predicting* the properties of the fabricated object for a given design. This has been called *forward design* and it replaces the costly and time consuming design cycle. A user knows in advance the mapping from design attributes to resulting properties, which allows extremely fast iterations without the need for physical realizations. Forward design involves building a computational model of the underlying process governing the properties of the object for simulation (e.g., mechanical elasticity for elastic deformation).

3. *Inferring* the design of the object that produces some target user-defined properties. This is the opposite functionality to forward design and has been called *inverse design*. This feature aims to replace –or at least reduce the need for– expert knowledge. The user is not required to know the details of the design, but only to specify the properties that the fabricated object should fulfill (e.g., balance, stability or buoyancy). Through numerical optimization of the computational model built for prediction, the computer infers the corresponding geometry and material attributes that give rise to user specification.

4. *Guiding* the design process. In some design problems, it is not even clear which is the *solution space*, i.e., the range of properties that can be achieved by fabricable objects. Understanding of the solution space may be achieved by finding high-level descriptors of fabricable properties, so that the user can easily control them to define targets for inference. Nevertheless, more sophisticated implementations may directly allow designers to virtually navigate only the space of fabricable solutions, and even suggest alternative paths to spur the user's creativity.

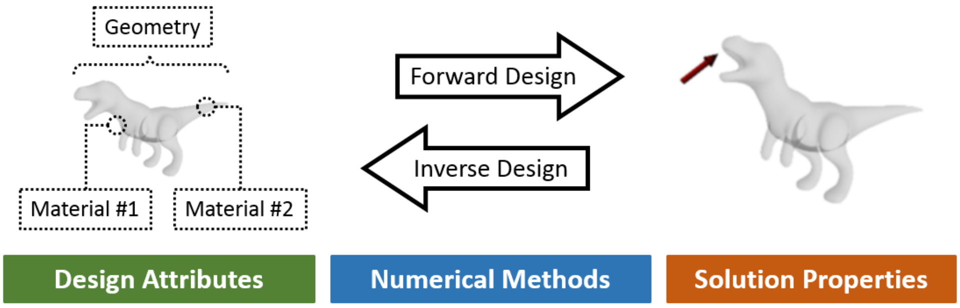


Figure 1.1: Modern computational design solutions provide computational models to represent design attributes and fabricated object properties, and implement numerical methods allowing the user to navigate both spaces through forward and inverse design.

Fabrication-oriented design tools proposed during the last few years feature some or all of these functionalities. We have witnessed computational methods for designing objects that can stand on their own [1], spin stably [2] or sound in a specific manner [3]. It is also possible to create mechanical characters capable of interesting motions [4] and produce 3D-printable robotic creatures [5] or telescoping structures [6]. And the list goes on. This rich environment creates fertile ground for creativity which will go even further in the future, when computational fabrication technologies become progressively more accessible to the public.

Overview

In this thesis, we deal with the computational design and automatic fabrication of flexible structures: assemblies of –potentially heterogeneous– interrelated elements that exhibit elastic behavior. With the development of these methods, we contribute on each of the functionalities mentioned above: i) we build compact parameterizations of complex structures composed of elastic rods and fabrics; ii) we devise mechanical models that allow the prediction of their deformation behavior; iii) we create optimization methods that automatically infer fabricable designs from user-defined functional goals; and iv) we propose interaction approaches that empower the user with further understanding and facilitate navigation of the solution space.

In the following sections, we give a brief overview on these topics, introducing the scientific and technical challenges that we have addressed, and reviewing the contributions that we have developed.

1.2 Designing flexible structures

The design and fabrication of deformable objects has been an integral part of human history since the origins of clothing. The adaptability of deformable materials allows compliant tools to remain functional in dynamic and constrained environments, and to safely interact with life forms and fragile objects. Soft materials are being used increasingly more often for the production of toys, apparel, furniture, architecture or robots, among others.

Originally limited to rigid materials, computational fabrication methods have recently opened the door for fast prototyping of deformable objects. While rigid designs merely require the specification of the geometry, the behavior of a deformable object is also heavily influenced by the spatial distribution of material attributes. This poses a grand challenge on computer-aided design as it is non-trivial to model the continuum mechanics that drive the relationship between the local combination of geometry and materials, and the corresponding global deformation behavior.

This topic has received increased attention from the computer graphics community over the last few years. Most of the works differentiate from each other in how they define the computational model that represents the design space. This characteristic is typically determined by the fabrication technology that is being employed. For instance, some works use a discrete combination of template materials with different deformation behaviors [7]. Other works focus on changing the overall shape of the objects to determine their deformation properties [8, 9]. Multi-material printing technologies allow us to consider an heterogeneous spatial distribution of material attributes [10, 11]. Nevertheless, several recent works are based on defining small-scale geometry attributes to overcome the use of a single base material [12, 13]. Alternatively, other works have focused on creating interactive solutions that allow the user to rapidly navigate the space of fabricable designs [14, 15, 16]

The design and fabrication methods developed in this thesis are in line with the two latter groups of works. We aim for reducing fabrication complexity by using a single material, and we rely on local geometry changes to determine the deformation behavior of the flexible structures. The interrelation between the components of the assembly poses further challenges for the definition of the design space, which must be compact, tractable, expressive, and it must naturally produce coherent designs. At the same time, we favor approaches that enable the designers to define the properties of the fabricated objects either using high-level descriptors or interactive techniques. The latter imposes and additional challenge in terms of performance.

Forward and inverse design functionalities heavily rely on mechanical simulation and elastic shape design through numerical optimization. These are the two foremost technical challenges that must be overcome for the computational design of flexible structures.

1.2.1 Mechanical simulation

The computational design of deformable objects relies on mechanical simulation to accurately predict which will be the static deformation properties of a given design once fabricated. The creation of analytical models to explain the behavior of real-world phenomena has been traditionally addressed by classical physics. Engineering fields have been devoted to the development of computational model and numerical methods that allow computers to predict the behavior of materials and structures. In the last few decades, since the pioneering work by Terzopoulos et al. [17], computer graphics has heavily contributed to the innovation in this area motivated by the creation of realistic virtual representations of real-world phenomena.

Our work deals with the design and fabrication of flexible structures composed of elastic rods and fabrics. The accurate simulation of the elastic behavior of fabric has been successfully tackled in the past [18, 19]. Similarly, several discrete elastic models have been suggested for the simulation of individual rods [20, 21]. However, modeling the deformation behavior of an assembly of elements, possibly of a diverse nature, has not been addressed so often: the structure should remain consistent and the different components should interact with each other adequately, i.e., point forces and rotational torques are transferred. In this thesis, we develop simple yet accurate computational models for the simulation of elastic rod meshes [22], in chapter 5, and tensile structures [23], in chapter 6. In both cases, we aim for creating solutions that satisfy two main requirements:

- Given some material characterization, simulations are experimentally validated to ensure that models produce accurate predictions of the elastic behavior of the structure, more precisely, its shape in static equilibrium.
- Simulations are efficient and avoid using numerical constraints for the coupling between the different elements. This is preferable so that the computational model lends itself well to numerical optimization, as this is necessary for inverse design.

Although developed in the context of a computer-aided design tool, these computational models might be of great use in other scenarios where an efficient yet accurate simulation is needed, e.g., animation or video games. In chapter 3, we review the foundations of mechanical simulation and provide a detailed explanation of all the models used for the development of this thesis, including our own contributions.

1.2.2 Inverse elastic design

Our tools feature inverse design operators that automatically determine attributes of the elastic structures such that their deformed shapes in static equilibrium meet some user-specified goal. Solving this problem constitutes a very hard challenge for two reasons:

1. Any feasible deformed shape of the structure must be in static equilibrium. This imposes a hard constraint on the solution space and limits the satisfaction of user-specified goals, which will be infeasible in most cases.
2. Similarly, the fabrication technology imposes bounds on the design space, as often it is not possible to use arbitrary fabrication materials or arbitrarily complex geometry.

The associated constrained optimization problem heavily relies on mechanical simulation and has been repeatedly formulated throughout the extensive literature on computational design of deformable objects [7, 14, 10, 16, 24, 25]. Standard numerical optimization methods could be used to solve such problems, but they often do not guarantee finding a good solution. Together with an appropriate definition of the design and solution spaces, challenges come with the selection of the particular optimization approach. In this thesis, we adopt an approach based on the iterative navigation of the equilibrium constraint manifold, similar to [14, 26]. Built upon this idea, we make additional contributions to deal with two problems:

1. **Local minimum solutions**. In the context of flexible rod meshes (chapter 5), we identify the attributes of the design space having the largest effect on the reduction of the bulk fitting error, and we propose a multi-resolution algorithm that incrementally considers additional design parameters using interpolation.
2. **Highly constrained problems**. In the context of Kirchhoff-Plateau surfaces (chapter 6), we propose a two-step algorithm to interactively explore the solution space of highly constrained problems, for which user-specified goals are generally too far from being feasible.

Optimization challenges similar to ours are common across many other design problems, and other authors might find inspiration in our solutions to face their own challenges. In chapter 4, we formally characterize our shape optimization problem, and we discuss other alternative solving approaches.

1.3 Flexible rod structures

Rods are curve-like elastic bodies that have one dimension (”length”) much larger than the others (”section”). Assemblies of rods are extensively used in structure engineering, where rigid truss structures are specially valued for their light weight, high stiffness and low cost in terms of transport, storage and assembly time [27, 28]. Their flexible counterpart are called *active-bending structures*, and appear in combination with elastic membranes for the creation of tensile structures that provide more design freedom and reduce the amount of external supports needed [29, 30]. The use of rod-like structures has been mainly restricted so far to architectural geometry applications, and has just started to receive attention from the computer graphics community for more general design applications like minimizing printing material [31] or producing wire meshes [32], reciprocal frames [33] and tensegrities [34], among others.

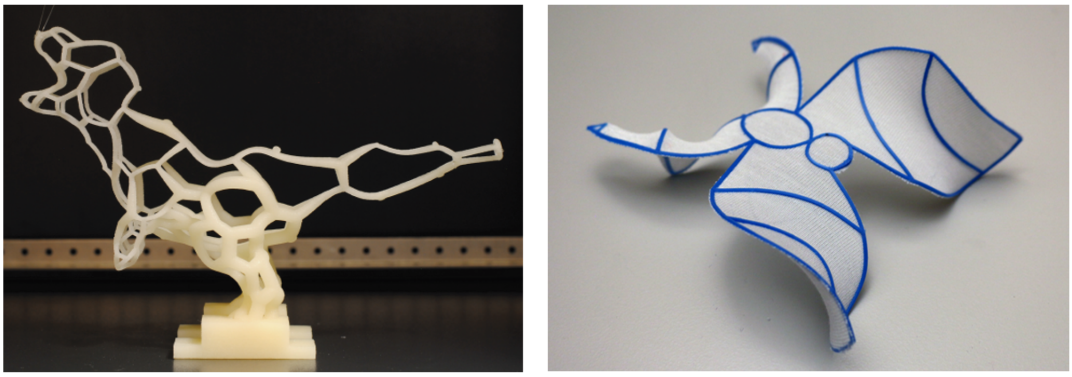


Figure 1.2: In this thesis, we explore the computational design and fabrication of two instances of elastic rod assemblies: flexible rod meshes (left) and Kirchhoff-Plateau surfaces (right).

Nevertheless, these structures are still underused and, as we demonstrate in this thesis, might find many other potential applications. We validate our computational tools with the design and fabrication of two different instances of flexible rod structures: *flexible rod meshes* and *Kirchhoff-Plateau surfaces* (Fig. 1.2).

1.3.1 Flexible rod meshes

Flexible rod meshes are assemblies of elastic rods. These structures are lightweight, relatively low-cost and can also form the support structure of solid objects if filled or dressed with other materials. But most importantly, the global deformation properties of a rod mesh can be adjusted simply by locally varying the cross-sectional properties of the rods, i.e., the radii. Consequently, a heterogeneous deformable object can be fabricated in one piece and from a single base material. This constitutes a key feature in the current scenario of low-cost consumer level printing machines.

In chapter 5, we consider flexible rod meshes as an implementation of deformable surfaces and push the limits of their expressive power. Our computational tool takes as input several poses of a deformable surface with known boundary conditions and automatically computes the rod mesh that best approximates the desired shapes. From several deformation instances with different elasticity properties, we seek a unique design that generalizes all the behaviors. From the design perspective, this facilitates producing complex deformation features like anisotropy, heterogeneity and model merging. In order to solve the associated optimization problem, we propose a simulation-based approach where the selection of the design space and the definition of a proper optimization strategy are essential.

In this thesis, we have explored the potential of flexible rod meshes for toy and apparel design, but they might also find potential application in other fields like furniture design, soft-robotics, orthotics or wearable assistance.

1.3.2 Kirchhoff-Plateau surfaces

Kirchhoff-Plateau surfaces (KPS) are planar rod meshes embedded in pre-stretched fabric that deploy into complex three-dimensional shapes. In their deformed state, these structures consist of a combination of piece-wise minimal surface patches, i.e., patches shaped as the surface of minimum area that spans a given boundary. Furthermore, such boundaries can only assume shapes corresponding to the static equilibrium of a planar rod mesh under membrane tension. These seemingly complex shapes may attract the attention from the design and research communities for two main reasons:

- From an application point of view, minimal surfaces are appreciated for their smooth aesthetic appearance and inherent material efficiency and structural stability. Minimal surfaces are widely used for light-weight and cost-efficient structures, ranging from large-scale roofs, canopies and shade systems, to acoustic deflectors, light diffusers and decorative elements for interior design.

- From a theoretical point of view, at small scale, these structures can be easily manufactured by 3D-printing planar rods onto stretched fabric, as recently demonstrated by [35]. Being able to easily design and manufacture instances of the Kirchhoff-Plateau problem may intrigue theorists and designers that have struggled with producing physical realizations.

From a design perspective, considering the highly constrained solution space, we cannot expect that there exists a design that closely approximates a user-defined target shape, but nevertheless KPS provides ample room for shape abstraction, interpretation and creativity.

In chapter 6, we turn away from fully-automated solutions in favor of a user-guided but computer-assisted design paradigm. Here the user is responsible for creating the topology of the rod mesh and for transforming it into the desired shape using a set of modeling tools that implement editing operations directly on the equilibrium state of the surface. We highlight a two-step optimization scheme which is essential for the implementation of the inverse design functionality, which allows the designer to effectively navigate the space of feasible solutions interactively. We demonstrate our method by designing a diverse set of complex-shaped KPS, each validated by a physically fabricated prototype.

1.4 Contributions and publications

The main contributions of this thesis are the following:

- A mechanical model for the accurate and efficient simulation of flexible rod meshes that implicitly handles coupling between rods without requiring numerical constraints or any additional degrees-of-freedom. (Chapter 3)
- A computational model for the representation of the design space of flexible rod meshes that is compact, expressive and naturally produces good quality designs through Hermite interpolation. (Chapter 5)
- A multi-objective sensitivity-based optimization method for the inverse elastic design of flexible rod meshes. In combination with the previously mentioned mechanical model and design space definition, it allows us to infer the design attributes of a rod mesh such that its static equilibrium matches an arbitrary number of user-defined goals with specific boundary conditions. (Chapter 5)
- A multi-resolution optimization strategy for improving the convergence of inverse elastic design problems, which adaptively increments the detail level of the design space parameterization when needed. (Chapter 5)
- A computational model for the compact representation of the design space of Kirchhoff-Plateau surfaces that naturally handles coupling between rods and fabric using collocation and Laplacian interpolation. (Chapter 6)
- A complete computational tool for the design of Kirchhoff-Plateau surfaces that features multiple editing operations affecting both the design and solution spaces. Such editing operations include: i) forward geometry and topology

editing, with real-time feedback on the resulting deformation, ii) sensitivity-based modal exploration of the solution space, iii) interactive inverse modelling of the equilibrium shape with design constraint preservation, and iv) visualization of mechanical properties for decision-making. (Chapter 6)

- A two-step optimization strategy based on incrementally exploring an approximation to the solution space in order to define close-to-feasible target deformations for the inverse elastic design problem. (Chapter 6)

The results corresponding to the contributions of this thesis are compiled in the following two works published as a first author in ACM Transactions on Graphics, and presented at the ACM SIGGRAPH conference:

- Jesús Pérez, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, José A. Canabal, Robert Sumner, and Miguel A. Otaduy. 2015. Design and fabrication of flexible rod meshes. *ACM Trans. Graph.* 34, 4, Article 138 (July 2015).
- Jesús Pérez, Miguel A. Otaduy, and Bernhard Thomaszewski. 2017. Computational design and automated fabrication of Kirchhoff-Plateau surfaces. *ACM Trans. Graph.* 36, 4, Article 62 (July 2017).

1.5 Outline

The rest of the thesis if organized as follows:

- **Related work**. Chapter 2 reviews the main works that have inspired and influenced the development of this thesis, grouping the extensive literature in two general themes: mechanical simulation and computational design.
- **Mechanical simulation**. Chapter 3 overviews the relevant foundations of mechanical simulation and describes in detail the mechanical models and numerical solvers that have been developed in this thesis, focusing on rod-mesh and thin-shell mechanics and the static equilibrium computation.
- **Inverse elastic design**. Chapter 4 presents a formal characterization of the inverse elastic design problem, describes some numerical solving methods, and analyzes the potential problems of their practical implementation, focusing on sensitivity-based constraint exploration.
- **Flexible rod meshes**. Chapter 5 focuses on the computational design and fabrication of flexible rod meshes and describes the computational model and optimization methods that facilitate the solution of the inverse design problem.

- **Kirchhoff-Plateau surfaces**. Chapter 6 explores the computational design and fabrication of Kirchhoff-Plateau surfaces and proposes a user-guided but computer-assisted tool that allows a user to interactively design such structures.

- **Conclusions**. Chapter 7 contains a discussion on the limitations of the methods presented in this thesis, analyzing their potential impact and suggesting possible future work.

Chapter 2

Related work

The design and fabrication of flexible structures is an interdisciplinary subject integrating knowledge across several related fields. This chapter reviews the main works that have inspired and influenced the development of this thesis. We group this extensive literature in two general themes: mechanical simulation and computational design.

- The first section 2.1 reviews works dealing with the creation of computational methods for the simulation of mechanical systems. We focus on mechanical modeling of elastic rods and thin shells, model coupling and numerical solving.
- The second section 2.2 groups many heterogeneous works related with computational design. We first categorize the literature according to several criteria and briefly review a few topics that are tangentially related with this thesis: surface design, animation control, material characterization and exploration of constrained spaces. We then focus on fabrication-oriented design problems with special emphasis on rod structures and physical surfaces.

2.1 Mechanical simulation

Classical physics has traditionally addressed the creation of analytical models to explain the behavior of real-world phenomena. From an engineering perspective, mechanical simulation comprises the development of computational models and numerical methods that allow computers to predict the behavior of materials and structures. In the last few decades, since the pioneering work by Terzopoulos et al. [17], computer graphics has heavily contributed to the innovation in this area due to its never-ending pursuit of more realistic virtual representations of the real world.

This pursuit is motivated by different purposes. Fabrication-oriented computational design, for example, often relies on mechanical simulation to accurately predict how a given prototype will behave once fabricated. Our work deals with the design and fabrication of deformable structures composed of elastic rods and fabric. In the following, we will summarize the main computational models that have been used for the simulation of elastic rods and thin shells and review the state of the art. Then, we briefly go over algorithms currently used to solve numerical integration problems derived from the partial differential equations of mechanics. Finally, we will introduce a few more works dealing with the combination of potentially heterogeneous models for the simulation of complex structure assemblies.

2.1.1 Elastic rod simulation

Rods are curve-like elastic bodies that have one dimension (”length”) much larger than the others (”section”). A great variety of rod simulation methods have been presented in the literature. Many, including the earliest approaches, deal with the efficient simulation of hair, highly demanded in the VFX and video games industries. These solutions are mostly based on mass-spring systems which explicitly represent the curve as a sequence of nodes joined together by elastic springs [36, 37, 38, 39]. While these methods are capable of modeling resistance to stretch and bending, their formulations do not come from the discretization of a continuous rod elasticity model. Plus, they are limited in their ability to model twist, anisotropy and curls.

Alternatively, framed representations describe the configuration of a rod by an adapted framed curve. The assignment of a material frame to each point on the centerline contains the requisite information for measuring the orientation of the rod section and hence enables advanced features. The corresponding governing equations –a set of partial differential equations together with boundary conditions– were first developed by Kirchhoff and Clebsch [40] in their theory of thin elastic rods under finite displacements. Some works following this idea solve the statics and dynamics of Kirchhoff rods using an explicit centerline representation based on the discretization of the Cosserat rod geometry model [41, 42, 20]. Other works however use an implicit centerline representation and introduce reduced-coordinates models based on a minimal parameterization needed to account for the exact kinematics for the rods. Examples of these works are the solutions based on articulated rigid body systems [43] or the super-helix model [44]. Relying on the Bishop frame, *discrete elastic rods* (DER) and *discrete viscous threads* (DVT) by Bergou et al. [45, 46] use a curve-angle parameterization to reduce the number of redundant parameters and guarantee that the orientation frame naturally remains adapted to the centerline. The latter approach constitutes the best starting point for our work, as it is experimentally validated, and it lends itself well to numerical optimization, as we demonstrate later in this thesis.

Rod meshes are networks of elastic rods that are linked by joints. Not many works before our own have addressed the simulation of this kind of structures. In general, previous approaches are based on the use of numerical constraints during simulation to ensure rods remain connected. Such is the solution by Bergou et al. [45], which uses bodies rigidly attached to rods as joints. Elastic joints were considered in *Cosserat Nets* by Spillman and Teschner [47], which extends the Cosserat model to branched and looped topologies albeit at the expense of also using constraints. More recent works have taken the *position-based dynamics* (PBD) approach introduced in [48] and adapted it to the simulation of elastic rod networks [49]. However, the lack of a formal connection between the positional constraints used in PBD and elasticity theory restricts the use of this solution to applications where a predictive representation is not needed. As it will be further explained in this thesis, the work presented in chapter 5 extends the DER model introducing a physically-based elastic energy at connections. This allows the accurate simulation of rod mesh mechanics without the need of numerical constraints, which reduce the complexity of our optimization problem. A similar formulation was suggested later in the work by Zehnder et al. [50] by adding rotational DoFs at each joint.

2.1.2 Thin-shell simulation

Thin shells are thin flexible structures with a high ratio of width to thickness (e.g. >100). Thin shells have been also extensively studied by the computer graphics community, focusing mainly on the efficient and accurate modeling and simulation of cloth mechanics. Cloth has exemplified advances in physics-based animation dealing with a broad range of problems such as mechanical modeling, contact handling, friction, plasticity, fracture and numerical integration, among others. In the following, we will roughly review the most important works and mention a few current hot topics. For a broad survey on cloth simulation we refer the reader to [18, 19].

Since the pioneering work on elastically deformable models by [51], the most widely used simulation techniques were initially mass-spring systems [52] and more general particle systems [53, 54]. The popularity of mass-spring systems is due to the ease of implementation and low computational cost, but the accuracy offered by this method is rather limited. As an alternative, continuum-based approaches lead to a set of partial differential equations which have to be discretized in space, generally using the *finite element method* (FEM). Most of the existing FE-approaches are based on the geometrically exact thin-shells formulation by Simo et al. [55]. With this basic continuum formulation, simple isotropic materials can be simulated consistently with reduced dependency on the specific discretization. Further works incorporated novelties allowing to model increasingly more complex behaviors that are observable on real-world textile materials such as anisotropy [56, 57, 58] and nonlinear deformations [59, 60, 58]. As happened to rod simulation, many current works are based on position-based approaches such as [61, 62] but are not accurate enough for fabrication-oriented design applications. Our work relies in the widely known *discrete shells* formulation [63] for the flexural energy and the classic *orthotropic St. Venant-Kirchhoff* constitutive model for the membrane energy (see e.g. [64]).

Apart from the mechanical model, during the last few years, there have been contributions in many other aspects of thin-shell simulation. To mention just a few examples, some works have tried to overcome the limitations of spatial discretizations using remeshing methods [65], which have been successfully applied to model complex phenomena such as folding [66] and fracture [67]. Other works have addressed the problem of modeling and estimating cloth hysteresis as an effect of the internal friction between yarns [68]. The augmentation of coarse cloth simulations with realistic-looking wrinkles have also attracted considerable attention and clustered a variety of solutions like subspace simulation using adaptive bases [69] or procedural wrinkle creation based on the coarse strain tensor [70, 71]. Finally, a promising research line related to both thin shells and rod mechanics proposes the simulation of cloth at the yarn level [72, 73, 74, 75], which allows showing new interesting effects and unprecedented levels of agreement to real-world materials.

2.1.3 Model coupling

This thesis deals with the computational design of structures composed by an assembly of objects, possibly of a diverse nature. To accurately predict their deformation behavior, it would be theoretically possible to use a single volumetric mechanical model. However, to obtain precise enough results, that would require to employing a very fine spatial discretization to effectively capture sufficient geometry detail at each individual object scale. The resulting computational problem would be intractable in most of the cases. As an alternative solution, computer graphics research has often considered the use of different mechanical models, conveniently selected to take advantage of the specific kinematic and mechanical properties of each individual object. For example, a rod mesh can be simulated as an assembly of elastic rod models, as in chapter 5, or use a cloth model coupled to a rod mesh model at specific locations to represent a tensile structure, as in chapter 6.

To guarantee that the structure remains consistent and the different objects interact with each other adequately, several coupling approaches have been explored over the years. The simplest solution is the collocation of DoFs: all objects attached to a given coupling area share their corresponding discretization variables. This ensures point forces are shared and the structure remains consistent in any case. However, rotational torques are not transferred and so this approach does not offer a complete solution for the interaction between parts. In the real world, structural joints are not just single points but volumetric components and, consequently, rigid bodies are often placed at connections to mimic this fact. In some computer graphics works, these joints have been explicitly represented with rotational DoFs [21, 50, 23], while in others, the rotation of the joint is implicitly derived from the state of all the attached elements [47, 22]. To ensure rotational torques are transferred, the attached components must remain aligned with the rigid body at the connection point. The straightforward solution is to employ soft or hard numerical constraints to enforce this condition, as in e.g. [45, 47, 49]. In general, this approach has been extensively used for modeling many phenomena in which any kind of continued attachment between objects takes place, for instance, in frictional [76] and adhesive [77] contact. An alternative to using constraints is the definition of physically-based energies, whose derivation leads to elastic coupling forces [78, 22, 50, 23]. The resulting joints are compliant, what, with an adequate material characterization, produces a better agreement with the behavior of real-world structural joints, even under coarse discretizations.

Nevertheless, model coupling is not only restricted to simple inertia transfer. Many interesting effects have been achieved over the last few years by studying how objects in a heterogeneous assembly interact with each other. For instance, a few works have developed coupling methods between thin shells and deformable solids that allow the simulation of skin sliding on top of the muscles [79], facilitate adding high resolution wrinkles to coarse volumetric models [80], or model friction and air effects between cloth and bodies [81]. In the context of model coupling, contact and sliding have been an interesting convergence point for several works built around the *Eulerian-on-Lagrangian* idea [82]. This method proposes the combination of the two simulation methodologies in mechanics, i.e. Eulerian and Lagrangian. This approach has been applied to formulate frictional contact models between rigid bodies and strands [83] or cloth [84] that preserve contact contours independently of the discretization level. A particularly interesting research line proposes the simulation of cloth at the yarn level [74, 85, 75]. This approach assumes all yarns share Lagrangian DoFs at their crossing points but are allowed to slide with respect to each other using Eulerian coordinates. Consequently, there is no need to explicitly handle contacts between yarns, and hence the simulation of large garments at the yarn level is more tractable. In our case, the same methodology might be applied in the future for the design of tensile structures with sliding components.

2.1.4 Numerical solving

Since the seminal work of Baraff et al [86], implicit methods have been the predominant choice in computer graphics for rod and thin-shell dynamics simulation –and in general for most physically-based animation problems. For a detailed overview and comparison of existing integration schemes and their efficiency –applied to cloth simulation–, the reader is referred to [87]. In our work, we have adapted the variational form of implicit Euler integration in [88] to solve dynamics, as well as the corresponding Lagrangian mechanics formulation of the static equilibrium problem. Recent approaches however have shifted towards position-based methods inspired by works like the previously mentioned PBD [48] and *projective dynamics* by Bouaziz et al. [89]. Initial implementations used positional constraints with no connection to classical elasticity theory, e.g. [61, 90, 91, 92]. However, more recent solutions like the one by Liu et al. [93] have formalized the connection between projective dynamics and quasic-Newton methods, allowing to leverage ideas from numerical optimization and thus support real-time simulation of many hyperelastic materials.

2.2 Computational design

From a general perspective, computational design deals with the creation of software that facilitate design tasks. It roughly comprises the development of a parameterized computational model of the designed entity, which allows computers to predict how this entity would perform considering aesthetics, functionality, efficiency and many other criteria.

This thesis mainly focuses on the design of flexible structures, i.e., finding the parameters of a mechanical model such that the resulting fabricated object behaves and/or looks in a desired manner. However, computer graphics has explored many other related design problems which have heavily influenced our own work. To give the reader an overall view of the diversity in the literature, we will categorize these works according to several criteria:

- **The nature of the underlying mathematical formulation**. Our own work together with many others [7, 14, 8, 16, 22, 24, 23, 94, 95, 25] relies on classical elasticity theory to predict the behavior of the fabricated objects. However, over the years, many works have also addressed kinematic [96, 97, 98, 5, 99], geometric [100, 101, 102, 103, 104, 105, 106] or light-interaction [107, 108, 109, 110, 111] design problems.

- **The definition of the design goal**. Our work belongs to a family that focuses on functionality and fabricability [14, 88, 97, 16, 22, 24, 23, 94] but many other works address questions like appearance and aesthetics [108, 112, 113, 50, 114, 111, 32] or manufacturing and material cost-efficiency [31, 115, 116, 117].

- **The solution approach**. Some works on computational design consider mostly forward solutions [14, 118, 30, 29, 119, 50]: the user iteratively modifies the parameters of the model while the computer provides a prediction of the resulting performance. More recent solutions have also explored inverse approaches [120, 1, 10, 2, 12, 22]: the user specifies the desired result and a numerical optimization algorithm outputs a valid solution such that the corresponding behavior is as close as possible to the goal. Most current works however propose a combination of the two [97, 8, 16, 5, 23, 25].

- **Interactivity and user interaction**. Methods can be classified into those that do require user interaction [14, 8, 16, 97, 16, 5, 25, 23], and those that do not [88, 1, 10, 2, 12, 22]. It is also important to highlight those methods that provide output feedback at interactive rates [26, 121, 31, 122, 15, 23, 123, 124].

Recently, Bermano et al. [125] presented a very thorough review of fabrication-aware design where works are also classified attending to several criteria such as object representation (e.g., surface, volume), designed attributes (e.g., shape, material) and goals (e.g., appearance, deformation, motion). We refer the reader to this work for further details. In the following pages we will focus on the the extensive literature on computational design that is most relevant to the work developed in this thesis. We will start going through some geometric problems focusing on surface design. Then, we explain several works dealing with animation control, material characterization and constrained space exploration, and how they are tangentially related to our problem. Finally, we will focus on fabrication-oriented design, grouping the extensive body of work according to how it is related to ours.

2.2.1 Surface design

Our work on KPS presented in chapter 6 targets the design of a class of physical surfaces whose shapes are governed by a particular set of equilibrium constraints. Surface design has been in the focus of computer graphics ever since its beginnings. Numerical problems associated with this works are similar to our formulation, although not all of this problems originate from mechanical simulation and they do not explicitly consider the fabricability of the designed surface.

For instance, *developable surfaces* have attracted a lot of attention from computer graphics [101, 102, 126, 127, 128]. They arise naturally when creating 3D surfaces from flat, inextensible material such as plastic, paper, or stiff fabric and so are closely related to fabrication-oriented design. As an example, based on the principle of auxetic materials, Konakovic et al. [129] were able to create doubly-curved surfaces by structuring planar sheets of quasi-inextensible material. This problem is also closely related with surface parameterization, which has many applications in computer graphics, for instance, for texture mapping. A good survey on the topic by Floater et al. can be found in [130]. While developable surfaces are characterized by having zero *Gaussian curvature*, requiring vanishing *mean curvature* leads to *minimal surfaces*.

Minimal surfaces have been intensively studied in mathematics [131, 100] and also computer graphics [103, 104, 105]. In the context of architecture and engineering, minimal surfaces appear naturally when designing tensile membrane structures; see, e.g., [132]. Beyond minimal surfaces and the related *Plateau problem*, there has been an increasing interest in the *generalized Plateau problem* which, instead of assuming a rigid boundary, considers the case of *Euler elastica* [133] and *Kirchhoff rods* [134]. While these works focus primarily on questions of existence and uniqueness, in our work [23], we investigate the problem of modeling such surfaces for the purpose of fabrication.

Apart from digital surfaces made for virtual worlds, one important physical application domain is architectural geometry [106]. Surfaces from this category are often subject to constraints relating, e.g., to the planarity of polygonal faces [135, 136] or to compression-only self-supporting structures [27, 137]. In addition to enforcing such constraints numerically, exploring the resulting design spaces is a challenging problem as well [26, 15]. One particular line of recent work [30, 29] has studied the forward design of membrane structures coupled with bending-active elements. However, to the best of our knowledge, the inverse problem of automatically determining parameters such that the resulting equilibrium shape approximates given design goals has not been investigated so far.

2.2.2 Animation control

Most current animation systems rely on physically-based methods for the realistic depiction of real-world materials such as hair, cloth or fluids. Experts working on video games and the VFX industry are responsible for tuning the corresponding simulation parameters so that the final behavior responds to some artistic intention. This might result in a very slow trial and error process. As a consequence, the problem of optimizing external forces, material parameters and rest shapes to achieve a certain mechanical behavior has also been explored in the context of animation.

As an example of early work, Kondo et al. [138] controlled the deformation of elastic objects by keyframing rest shapes. Very similar methods have been used for the same purpose focusing on other mechanical models. For cloth simulation, for instance, Twigg and Kacic-Alesic [139] computed rest lengths for mass-spring systems in order to achieve desired garment drapes under gravity. Mainly applied to hair simulation, the same problem has been investigated for a number of different curve and rod models, including 2D elastic curves [140], articulated rigid body chains [43] and the super helices rod model [141]. In the context of character animation, Coros et al. [142] show that it is possible to create autonomous characters modeled as elastic objects by optimizing their rest shapes as a function of high-level motion goals. More generally, the methods described by Martin et al. [88] and Schumacher et al. [143] are also related to the problem of modulating rest shapes in order to effect the behavior of dynamic simulations, but solve the problem by adding a potential energy term specified relative to a set of input example shapes. One particular line of work has focused on the interactive editing of deformable simulations using techniques such as space-time methods or rotation-strain reduced coordinates to efficiently handle the high computational load [121, 122]. Finally, as an alternative to rest-shape design, different material optimization solutions have also been proposed with the same purpose. These works employ a variety of techniques such as model reduction [144] and optimization of principal stretches [145].

2.2.3 Material characterization

Correctly predicting object deformation requires i) a computational model that is capable of representing the material behavior, and ii) an accurate estimation of the parameters of the fabrication material. For the development of this thesis, we had to estimate material parameters from data. Additionally, the mathematical formulation associated with this task is equivalent to the one of a design problem. Basically, we look for object attributes (in this case, material parameters) such that the predicted deformation matches a target observed behavior.

Some works in computer graphics have addressed the data-driven estimation of material parameters specially in the context of computational cloth. Early ones [53, 54] were based on fitting parameterized functions of the deformation to measured data, extracted from comprehensive sets of experiments with their corresponding machinery –e.g., the *Kawabata evaluation system* [146]. Later, a few works improved previous results by measuring complex 3D deformations instead of just considering one-dimensional force-displacement curves. For instance, in the work by Wang et al. [147], a piecewise linear elastic cloth model is proposed and its parameters are fitted to experimentally acquired data using a simple setup. On the contrary, Miguel et al. [148] proposed a more powerful and sophisticated solution where nonlinear stress-strain curves are numerically optimized considering also the loading forces over boundary conditions.

In an alternative line of research, some works have proposed more inexpensive acquisition processes, like extracting parameters from casually captured videos [149, 150, 151]. Their aim was to avoid the need for controlled conditions, but as a consequence, it was not possible to separate internal (i.e., material-specific) from external (e.g., friction, air drag) parameters. More recent works have precisely focused on modeling such phenomena and trying to estimate their corresponding parameters. A pair of examples are the work by Miguel et al. [68] on internal friction and the one by Xu et al. [152] on damping.

2.2.4 Exploration of constrained spaces

Many of the previous works feature interactive applications and require the user to participate –at least– in part of the design process, e.g. [97, 8, 153, 154, 124, 123]. However, part of this literature that deals with highly constrained problems has focused more on specifically providing methods for the exploration of such spaces.

Many works on constrained geometric modeling are based on using optimization methods to minimize the distance from a constrained design to a target shape. This is the case of the work by Bouaziz et al [155], where a shape proximity function and projection operators are defined to perform constraint-preserving changes on arbitrary geometric data sets such as curves, polygons or volume elements. Another case is [26], which explored such constrained manifolds through first- and second-order approximations and applied this method for the design of planar quad meshes. In addition to allowing manual specification of target shapes, authors often leverage modal analysis to directly explore feasible solutions parametrically close to a given design. Freeform architectural design has made good use of early constrained mesh exploration approaches. For instance, Deng et al. [15] proposed a method where the numerical optimization is split into a sequence of simple subproblems that can be solved efficiently and accurately. Our own work on KPS (chapter 6) is greatly inspired by all these contributions, as we also propose a two-step optimization procedure based on a first-order approximation to the constraint, and we also feature modal analysis.

In general, all these methods assume that the user is responsible for the creation of the design topology and focus on the continuous optimization of low-level local geometry. However, there are some applications for which high-level structural features are also essential. Structure-aware shape processing addresses the problem by dealing with the global inter- and intra-semantic relations among the parts of a shape rather than on their local geometry. For an extensive review on this topic, we refer the reader to the work by Mitra et al. [156]. Structural relationships have been also studied in the context of different design problems. For instance, Guerrero et al. [119], propose a method to explore the variability of 2D geometric patterns, using different interpretations of their regularity that correspond to different design variations. The work by Umetani et al. [118] proposes an interactive solution for the guided exploration of physically valid shapes in furniture design. While the user focuses on the aesthetics, the framework helps to achieve physical realizability. This is done by generating multiple suggestions involving both discrete and continuous changes to restore validity when one or more constraints are violated. Generation of plausible furniture variations was further explored by Zheng et al. [157], where symmetric functional arrangements of substructures are identified, allowing the combination of parts across different model families. More recently, similar ideas have been applied to the creation of mechanical objects. In the work by [158], for instance, the authors present an interactive system that –using 3D models and high-level functional rules (e.g., A fits in B)– is able to optimize the shape geometry to produce a working design. In our works, the topology of the designs is either automatically generated, chapter 5, or delegated to the user, chapter 6. However, the implementation of a variation of some of these methods has been considered for future work.

2.2.5 Fabrication-oriented design

Designing fabricable objects whose properties can be intuitively specified and controlled is an important research challenge that is quickly gaining interest in the computer graphics community. As mentioned above, a great variety of performance criteria have been considered.

In the following, we will focus only on functionality goals in the form of motion and deformation behavior, as they are closer to our particular work. A variety of methods that investigate these aspects of design for rigid objects have been proposed. For example, there are computational design methods for creating objects that can stand on their own [1], spin stably [2] or sound under contact in a specific manner [3]. Thanks to recent works, it is also possible to create articulated 3D-printable representations of virtual characters [96, 159], design mechanical characters capable of interesting motions [97, 160, 4, 99] and produce 3D-printable robotic creatures [5] or telescoping structures [6]. Our work is closer to methods that control the deformation behavior of elastic objects –a problem that has received increased attention from the research community over the last few years. As also happened with animation control, some of these works dealt specifically with *material* design while others have focused more on the *shape* of the object.

- **Material design.** Current fabrication methods allow the use of multiple materials within a single printed object. Many works have taken this technology as the basis of their approach. For instance, Vidimče et al. [161] proposed a framework for the fabrication of objects composed of multiple materials and Bickel et al. [7] used a small set of template materials with different deformation behaviors to fabricate objects whose force-deformation response matches measurements of real objects. Similarly, Skouras et al. [10] described an optimization method for computing an inhomogeneous distribution of material parameters to control the way in which fabricated elastic objects deform under the influence of external forces.

- **Shape design.** The rest configuration of elastic objects can also be computed using automated methods. For instance, Skouras et al. [8, 16] showed that the shape of inflatable balloons and more complex structures can be controlled to match input targets by optimizing the rest state of membrane-based

> models. Similarly, the rest state of volumetric objects can be modified to control the deformation behavior of skin for robotic faces [120], or to specify the way in which elastic objects deform under gravity [9]. Many design problems based on single-material fabrication technologies require changing the shape of the object at a small scale to control deformation properties. Such is the case of the works developed in this thesis, where the stiffness of the rod structure is controlled by modifying rod radii. Microstructures is a related research line that has attracted increased attention over the last few years, see, e.g., [12, 162, 163, 13]. These works propose methods for fabricating deformable objects with spatially varying elasticity using 3D printing and a single relatively stiff material. They employ assemblies of small-scale structures with varied geometries that have an effect on global material compliance.

In either case, both material design and shape design problems reduce to identical mathematical formulations and can be treated in a generic manner. For instance, Chen et al. [164] presented a unified framework to fabricate objects with controllable deformation properties and Musialski et al. [165] proposed a series of numerical improvements that can be applied independently of the optimized features.

In general, the deformed shape of these elastic structures is governed by equilibrium conditions, thus requiring a balance between internal –elastic– forces and external forces such as self-weight and applied loads. The problem of designing deformable objects that assume desired equilibrium shapes under gravity has been extensively studied in computer graphics, e.g. for hair [140] and cloth [139] animation. In the context of fabrication-oriented design, similar problems have been addressed applied, for instance, to self-supporting surfaces [166] and custom-shaped elastic solids [9]. Our work shares many aspects of these inverse problems. However, our KPS design, chapter 6, differs in the sense that the driving force is also membrane stretch, which, unlike external loads, depends on the state of the system.

For completeness, let us briefly mention other performance criteria that have been considered in the literature. Some of these works have focused on appearance goals. For instance, Shuller et al. [114] proposed a unified framework to create surfaces that depict certain 3D shapes from prescribed view points. Consideration for aesthetics has also been seen in works dealing with the design of furniture [123] or ornamented decorative pieces [50]. Appearance goals are usually also combined with some high-level functionality. For instance, the work by Dumas et al [113] deals with the synthesis of structurally sound patterns from sample data, while Martinez et al. [112] use topology optimization to combine structural and appearance objectives into the same shape design. A completely different research line focuses on digital fabrication and deals instead with fabricability and material efficiency design goals. Some example works are the cost-effective printing of 3D objects with skin-frame structures by Wang et al. [31], or the more recent design of lightweight structures under force location uncertainty by Ulu et al. [117].

The following section finally reviews a few works dealing with two specific representations of fabricable objects which are specially related to the work developed in this thesis: physical surfaces and rod structures.

Physical Surface Design

Most works in computer graphics related with physical surface design have been restricted to flat panels, with applications to plush [167, 154], garment design [14, 168], rubber balloons [8], inflatable structures [16], thermo-formed models [114], surfaces composed of interlocking elements [153] and tensile structures [30, 29], among others. Most of them use a similar approach to the works developed in this thesis, with slight differences depending on the specific application.

For instance, rather than automatically generating a rod network for an input surface as we do in chapter 5, Skouras et al. [16] leave the topology problem to the user and instead optimize performance such as to enable fast design iterations. Our approach in chapter 6 follows the same spirit, but instead of optimizing for a fixed target shape, our method allows the user to explore the space of feasible designs using a set of editing tools that leverage simulation and optimization. This same paradigm of computer-assisted but user-guided design is implemented by several previous approaches, including the work by Umetani et al. [14], who use first-order sensitivity analysis in order to quickly predict the impact of parameter changes on the equilibrium shape of clothing. As one difference, our method extends this forward design approach with inverse modeling tools that, for user-specified editing objectives, automatically compute first-order optimal directions in parameter space. Allowing the user to directly edit the 3D equilibrium state was also the driving motivation for the work of Bartle et al. [168]. However, while their method uses a heuristics-based, gradient-free approach specifically tailored to the problem of pattern optimization for garment modeling, we capitalize on derivative information in order to implement fast forward and inverse design tools. Finally, we want to highlight the contemporary work by Guseinov et al. [94] on tension-actuated flat plates, which shares many similarities on the functionality goal to our own KPS design. While their solution allows to obtain doubly curved surfaces at the price of a more complex fabrication method, our results can be easily 3D printed but are restricted to piece-wise minimal surfaces.

Non planar surfaces mostly arise in architectural geometry, which is a very important application domain of computational design, specially in the context of freeform surface modelling. Although these approaches are not always directly applicable to real-world scales, most of these works feature fabricated samples to demonstrate the validity of their solutions. As a consequence, they share similar challenges to the ones described in this thesis. Generally, structures are mostly made of rigid materials and their design consider a combination of performance criteria including appearance, functionality and fabrication efficiency. For instance, the creation of surfaces composed of an assembly of polygonal faces has been extensively studied for their benefits in terms of material transport and storage [135, 136, 15]. Similarly, many works have dealt with the design of masonry structures, in which case, the low flexural strength of building mortar is countered using compression-only –self-supporting– designs [27, 137, 169]. For a more complete survey on this topic, we refer the reader to the review by Pottmann et al [106].

Rod Structure Design

Our rod meshes share some similarities with truss structures, which have been addressed in the context of digital fabrication. Wang et al. [31], for instance, proposed a computational design method that optimizes a truss structure to minimize the amount of printing material used in the fabrication of rigid objects. Constructions based on the same principle are known in architectural geometry literature as space structures and has also received attention from the computer graphics community, e.g., [28]. Apart from truss structures, more specific fabrication-oriented design problems have been considered over the last few years. For instance, Song et al. [33] proposed an interactive design tool for creating stable networks of interleaved rods (known as reciprocal frames), while Gauge et al. [170] addressed the design of physical characters using tensegrities –networks of rigid rods and elastic springs that are in static equilibrium.

Objects created by connecting and bending wires are also very common in furniture design, metal sculpting or wire jewelry, and have attracted increased attention over the last few years. For instance, Liu et al. [32] proposed a method for the image-based reconstruction of such structures, while Garg et al. [171] developed techniques for designing wire meshes with prescribed shapes. Wire meshes can be considered a special case of rod networks, but rather than physics-based optimization, their particular structures motivate a geometric approach. The approach by Miguel et al. [24] targets physical surface representations using bent wires that securely interlock by virtue of deformation. Our approach on KPS design shares the two-dimensional nature of their design space, but the underlying mechanics are very different. Instead of designing 3D networks that balance applied loads, our method computes a 2D layout optimized with respect to membrane forces.

While most of the above methods do not consider automatic topology generation, Zimmer et al. [172] focus exactly on this discrete problem, albeit in a context that does not involve equilibrium constraints. Similarly, the work by Zehnder et al. [50] faced the same problem and presented a computational tool for designing ornamental curve networks –structurally-sound surfaces with user-controlled aesthetics.

Chapter 3

Mechanical simulation

Our design tools build on physics-based animation in order to predict the deformation behavior of flexible structures in response to applied forces. In this chapter, we overview the relevant foundations of mechanical simulation and describe in detail the mechanical models and numerical solvers that we have developed in this thesis. Our main contribution here is the creation of a novel mechanical model for the simulation of flexible rod meshes in 3.2.2.

- The first section 3.1 briefly reviews the foundations of mechanical simulation, formalize the mathematical problem and establish a common framework for the definition of mechanical models.
- The second section 3.2 explains in detail the model that we have devised for the simulation of rod meshes, preceded by a description of the discrete elastic rods model upon which it is based.
- The third section 3.3 describes the discrete shells model that we use for the simulation of the tensile fabric, with special emphasis on the orthotropic constitutive model employed for the elastic membrane.
- The fourth section 3.4 concludes by explaining in detail the numerical methods that have been used for the computation of static equilibria and analyze potential problems of their practical implementation.

3.1 Foundations

Let us assume there exists a continuum mechanical model $\mathcal{M}(O)$ that determines the elastic behavior of a deformable object $O \subset \mathcal{R}^3$ through a conservative energy potential $V(\mathbf{O}) \in \mathcal{R}$. Simulating the statics of this mechanical system implies solving the nonlinear system of differential equations defined by net force equilibrium: $\mathbf{f} = \mathbf{0}$ for any applied external forces and boundary conditions. Finding a computational solution to these equations using simulation requires defining discrete expressions for both the kinematics of the object and the potential energy function.

Discrete kinematics We consider a generic Lagrangian discretization in the form of a vector $\mathbf{q} = \{q_1, \ldots, q_n\}$ in some known generalized coordinates system. This vector contains n independent variables that spatially approximate the state of the object in different configurations:

- We denote the *deformed* configuration of the object as $\mathbf{q}$ and refer to the space of all possible deformed configurations $\mathbf{q} \in \mathcal{R}^n$ as *deformed* or *world* space, Q.
- We denote the *undeformed* or *rest* configuration as $\bar{\mathbf{q}}$ and refer to the space of all undeformed configurations $\bar{\mathbf{q}} \in \mathcal{R}^n$ as *undeformed* or *material* space, U

Discrete energy Given a spatial discretization, an approximation of the potential energy must then be formulated, discretely parameterized by the deformed configuration $\mathbf{q}$, $V(\mathbf{q}) : \mathcal{R}^n \to \mathcal{R}$. Note that we are omitting here other magnitudes that might be also affecting the value of the energy potential (e.g. gravity constant, undeformed configuration material properties, etc.) but whose values do not change during the simulation. The discrete energy formulation allows deriving the expression of the system *forces* as the negative gradient of the potential energy V w.r.t the deformed configuration $\mathbf{q}$, $\mathbf{f}(\mathbf{q}) \in \mathcal{R}^n$:

$$\mathbf{f}(\mathbf{q}) = -\nabla_{\mathbf{q}} V^T = -\left(\frac{\partial V}{\partial q_1}, \ldots, \frac{\partial V}{\partial q_n}\right)^T . \tag{3.1}$$

Similarly, system forces *Jacobian* can be calculated as the matrix of negative second partial derivatives of the potential energy V w.r.t the deformed configuration $\mathbf{q}$, $\mathbf{J}(\mathbf{q}) \in \mathcal{R}^{n \times n}$:

$$\mathbf{J}(\mathbf{q}) = -\nabla^2_{\mathbf{q}} V = -\begin{bmatrix} \frac{\partial^2 V}{\partial q_{11}} & \cdots & \frac{\partial^2 V}{\partial q_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 V}{\partial q_{n1}} & \cdots & \frac{\partial^2 V}{\partial q_{nn}} \end{bmatrix} . \tag{3.2}$$

Note that, in our convention, we consider the partial derivatives of a scalar function to be a row-vector. With a close expression for both the force and the Jacobian of the mechanical model, the expression of the force can be linearly approximated at any point $\mathbf{q}_0$ using Taylor's expansion:

$$\mathbf{f}(\mathbf{q}) \simeq \mathbf{f}(\mathbf{q_0}) + \mathbf{J}\,(\mathbf{q} - \mathbf{q}_0) \tag{3.3}$$

Mechanical model With these ingredients, standard numerical methods can be used to solve the nonlinear systems of equations resulting from the static equilibrium. In the following sections, we will describe the different mechanical models we have used in this thesis in terms of these two components. First, we will overview the continuum mechanics formulation associated with the model. Then, a spatial discretization approach will be introduced followed by a reformulation of the smooth energies into discrete expressions.

3.2 Rod mesh model

Rods are curve-like elastic bodies that have one dimension (”length”) much larger than the others (”section”). As mentioned in chapter 2, among the many existing computational approaches for the simulation of rods, we choose the *discrete elastic rod* (DER) model by Bergou et al. [45, 46], due to its compact curve-angle representation with explicit centerline. We note, however, that other approaches based on reduced coordinates, e.g., [42], or full coordinates with constraints, e.g., [20], would be possible as well.

In the following section 3.2.1, we briefly summarize the most relevant part of the theory regarding Kirchhoff rods and discrete elastic rods. Since the model by Bergou et al. does not account for coupling among rods, in section 3.2.2, we propose a model for rod connections, formulate elastic energy terms, and describe how to correctly transmit bending and twist forces across connections.

3.2.1 Discrete rod model

In this section, we briefly review classic Kirchhoff rods theory as well as the reduced coordinate formulation of discrete Kirchhoff rods developed by Bergou et al. [21, 46]. We refer to the original works for a more thorough explanation.

Smooth Kirchhoff rods

Classic Kirchhoff rods are conveniently modeled as adapted framed curves with an arc-length parameterized centerline $\mathbf{c}(s) \in \mathcal{R}^3$ and an orthonormal material frame $\mathbf{m}(s) = [\mathbf{t}(s)\ \mathbf{b}(s)\ \mathbf{n}(s)] \in SO(3)$, where $\mathbf{t}$ is the tangent, $\mathbf{b}$, is the binormal and $\mathbf{n}$ is the normal. We assume that s is an arc-length material parameterization of the undeformed configuration of the adapted framed curve. Recall that we will denote quantities associated to the undeformed configuration of the material with an overline, e.g. $\bar{\mathbf{c}}$. The orthonormal material frame assigned to each point of the centerline spans the plane normal to the centerline's tangent called the cross-section, and contains the requisite information for measuring twist. Since the material frame is adapted to the centerline, the following condition must hold: $\mathbf{t}(s) = \mathbf{c}'/|\mathbf{c}'|$, where the prime indicates differentiation with respect to the s coordinate.

The Kirchhoff theory of elastic rods defines scalar functions that measure the strain of any adapted framed curve –given by the change of the orthonormal frame expressed in its own coordinates and the deformation gradient w.r.t. a naturally unitary straight rod:

$$\epsilon = \|\mathbf{c}'\|, \quad \omega_b = \boldsymbol{\kappa} \cdot \mathbf{b}, \quad \omega_n = \boldsymbol{\kappa} \cdot \mathbf{n}, \quad m = \mathbf{b}' \cdot \mathbf{n}, \tag{3.4}$$

where $\boldsymbol{\kappa} = \mathbf{t}'$ is the centerline's normal curvature vector. Here, the first term, ϵ, corresponds to the relative axial strain. The second and third terms, ω_b and ω_n, represent the rod's curvature vector expressed in material coordinates and measure the bending of the material frame. And the last term, m, refers to the twist of the material frame around the tangent. Accordingly, the total elastic energy contains stretching, bending and twisting contributions, $V = V_s + V_b + V_t$.

- The **stretch energy** is defined as

$$V_s = \frac{1}{2} \int k_s (\epsilon - \bar{\epsilon})^2 ds, \tag{3.5}$$

where k_s is the stretch stiffness.

- The **bending energy** term is defined as

$$V_b = \frac{1}{2} \int (\omega - \bar{\omega})^T \, \mathbf{B} \, (\omega - \bar{\omega}) ds, \tag{3.6}$$

where $\omega = (\omega_b, \omega_n)$ represents the centerline curvature vector and $\mathbf{B}$ is a symmetric positive definite 2×2 matrix representing the bending stiffness. This formulation allows for an anisotropic bending response in the two main directions of the cross-section.

- Finally the **twist energy** term is defined as

$$V_t = \frac{1}{2} \int \beta (m - \bar{m})^2 ds, \tag{3.7}$$

where β is the twist stiffness.

Note that this model generalizes to naturally non-unitary, curved or twisted rods by subtracting away the undeformed centerline axial strain $\bar{\epsilon}$, curvatures $\bar{\omega}$ and twist $\bar{m}$. The particular case of an isotropic, naturally straight rod is obtained by taking $\mathbf{B} = k_b \, \mathbf{I}_{2\times 2}$, $\bar{\omega} = \mathbf{0}$ and $\bar{m} = 0$.

Curve-angle formulation

The main contribution of Bergou et al. [21, 173] is the development of a reduced coordinate formulation of thin Kirchhoff rods, which is based on a previous analysis introduced by Langer and Singer [174]. In the curve-angle representation, the material frame is expressed through a rotation roll $\theta(s)$ from an adapted orthonormal reference frame $\underline{\mathbf{m}}(s) = [\underline{\mathbf{t}}(s) \; \underline{\mathbf{b}}(s) \; \underline{\mathbf{n}}(s)] \in SO(3)$ which will be further explained below. More precisely, material frame can be explicitly computed as $\mathbf{m} = \mathbf{R}(\underline{\mathbf{t}}, \theta) \, \underline{\mathbf{m}}$, where $\mathbf{R}$ is the rotation around the vector $\underline{\mathbf{t}}$ by an angle θ. When applied to an orthonormal frame, this operation reduces to:

$$\begin{aligned} \mathbf{b} &= \cos\theta \cdot \underline{\mathbf{b}} + \sin\theta \cdot \underline{\mathbf{n}}, \\ \mathbf{n} &= -\sin\theta \cdot \underline{\mathbf{n}} + \cos\theta \cdot \underline{\mathbf{b}}, \end{aligned} \tag{3.8}$$

with $\mathbf{t} = \underline{\mathbf{t}}$. Like other reduced coordinate models, this avoids the need for stiff constraints that couple the material frame to the centerline, while at the same time, the explicit centerline representation facilitates collision handling and rendering.

Discrete rod kinematics

In the discrete setting, the kinematic state of the rod results as follows: the centerline of the rod is represented by the centerline as a piece-wise linear curve defined by a set of n nodes $\mathbf{c} = \{\mathbf{c}_i\}$ and material frames are represented by a set of roll angles $\{\theta_i\}$ w.r.t. the reference frames that pertain to the edges of the centerline $\mathbf{e}_i = (\mathbf{c}_{i+1} - \mathbf{c}_i)$ (Fig. 3.1).

Whenever there is a change in the kinematic state of the rod, the reference frame $\underline{\mathbf{m}}$ should be updated so that the material frame $\mathbf{m}$ remains adapted. In discrete elastic rods, this operation is performed by parallel transport of a frame initialized to the material frame in the rest state. Parallel transport is an important concept in the discrete elastic rods model, which serves to transform a frame adapted to a source edge into a destination edge. Given two unit vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, the parallel transport is defined as the minimum rotation that aligns $\mathbf{v}_1$ with $\mathbf{v}_2$:

$$\mathbf{P}(\mathbf{v}_1, \mathbf{v}_2) = \mathbf{R}(\mathbf{v}_1 \times \mathbf{v}_2 / \|\mathbf{v}_1 \times \mathbf{v}_2\|, \angle(\mathbf{v}_1, \mathbf{v}_2)), \tag{3.9}$$

where $\mathbf{R}(\mathbf{v}, \psi)$ is the rotation about the unit vector $\mathbf{v}$ by an angle ψ, and $\angle(\mathbf{v}_1, \mathbf{v}_2)$

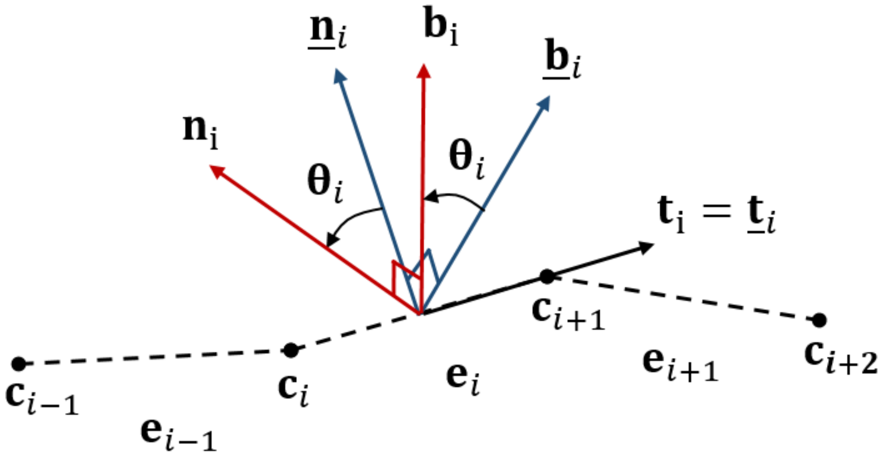


Figure 3.1: Discretization of the centerline as a piece-wise linear curve defined by a set of nodes $\mathbf{c}_i$ and edges $\mathbf{e}_i$. In the curve-angle representation, the material frame $\mathbf{m}_i$ (red) is defined as a rotation about the tangent of a reference frame $\underline{\mathbf{m}}_i$ (blue) by a roll angle θ_i.

is the angle between the vectors $\mathbf{v}_1$ and $\mathbf{v}_2$. Depending on whether reference frames are parallel transported in *space* or in *time*, the resulting formulation offers different computational advantages:

- **Parallel transport in space** Initially, Bergou et al. [21], suggested using the geometrically most natural (and physically most relaxed) frame adapted to a curve: the *Bishop frame*. For a given centerline, this is an adapted frame with zero twist uniformly, and its evolution along the centerline corresponds infinitesimally to a rotation about the curvature binormal. In the discrete setting, this implies parallel transporting frames between adjacent edges $\underline{\mathbf{m}}_k = \mathbf{P}(\mathbf{e}^t_{k-1}, \mathbf{e}^t_k)\,\underline{\mathbf{m}}_{k-1}$. During the simulation, the reference frame is subsequently updated starting from the frame at the beginning of the rod $\underline{\mathbf{m}}_0$. The state of the k-th material frame of the rod is dependent on the kinematics of the $(k-1)$ previous centerline edges. Consequently, the force stencil is not local and the resulting Jacobian is dense. Although this limits the applicability of this method, authors claim that, under specific conditions –i.e. isotropic bending response, straight undeformed configuration, quasistatic material frame and explicit integration– there is no computationally more efficient solution.

- **Parallel transport in time** Posteriorly, Bergou et al. [46], suggested that the reference frame would also evolve naturally in time, rotating by the minimum amount needed to keep itself adapted; this corresponds to a frame having zero tangential angular velocity. In the context of static equilibrium solving, “time” will refer to the subsequent iterations of the quasistatic solver. In the discrete setting, this is equivalent to parallel transporting the reference frame between two time-instances of the same edge $\underline{\mathbf{m}}_k^{t+1} = \mathbf{P}(\mathbf{e}_k^t, \mathbf{e}_k^{t+1})\, \underline{\mathbf{m}}_k^t$. During a simulation, this evolution can be done incrementally –only the reference frame at the current time step must be known to advance it to the next time step. The resulting force stencil is local and the Jacobian is sparse, enabling the use of implicit integration. Given the particular needs of our problem, we adopt the second approach.

With this reduced-coordinates formulation, the twist of the rod depends only explicitly on the set of twist angle coordinates. However, parallel transport establishes an implicit link between the kinematic state of the centerline and the twist angle, which must be considered during force and Jacobian derivation.

Discrete elastic energies

With the discretization, the elastic energies introduced before result:

1. **Stretch**. For an edge between nodes $\mathbf{c}_1$ and $\mathbf{c}_2$, with rest-length l_0, and stretch stiffness k, the stretch energy depends only on centerline variables and can be computed as follows:

$$V_s = \frac{1}{2}\, k_s\, l_0 \left(1 - \frac{\|\mathbf{c}_2 - \mathbf{c}_1\|}{l_0}\right)^2. \tag{3.10}$$

2. **Bending**. Recall that the elastic bending energy in the smooth case was dependent on the curvature of the centerline. Since each edge is straight, it follows that discrete curvature is associated with vertices. For two adjacent edges with material frames $\mathbf{m}_1 = [\mathbf{t}_1, \mathbf{b}_1, \mathbf{n}_1]$ and $\mathbf{m}_2 = [\mathbf{t}_2, \mathbf{b}_2, \mathbf{n}_2]$, the discrete curvature binormal at the shared vertex is a vector orthogonal to the osculating plane passing through the edges and can be computed as:

$$(\kappa \mathbf{b}) = \frac{2\mathbf{t}_1 \times \mathbf{t}_2}{1 + \mathbf{t}_1^T \mathbf{t}_2}. \tag{3.11}$$

It can be shown that the magnitude of this vector is $2\tan(\phi_i/2)$, with ϕ_i the turning angle between two consecutive edges. Note that this quantity is well-defined in the case of collinear edges even though the binormal is not. Given this, the discrete material curvature at the shared vertex is defined as

$$\boldsymbol{\kappa} = \left((\kappa \mathbf{b})^T \frac{\mathbf{n}_1 + \mathbf{n}_2}{2}, -(\kappa \mathbf{b})^T \frac{\mathbf{b}_1 + \mathbf{b}_2}{2}\right)^T. \tag{3.12}$$

With a rest-state curvature κ_0 and anisotropic bending stiffness $\mathbf{B}$ (see material parameterization below), the resulting discrete bending energy between the edges is

$$V_b = \frac{1}{2\,l_0}(\boldsymbol{\kappa} - \boldsymbol{\kappa}_0)^T \mathbf{B} (\boldsymbol{\kappa} - \boldsymbol{\kappa}_0). \tag{3.13}$$

It should be noted that $\boldsymbol{\kappa}$ is defined per vertex as a function of the incident edges and their material frames. Hence, κ depends on both, centerline positions and twist angles.

3. **Twist**. For the same two adjacent edges mentioned above, twist can be computed as the difference in the material-reference roll angle between the two frames, i.e., $\theta_2 - \theta_1$, plus a reference twist ψ introduced when parallel transporting the reference frame over time. For two subsequent steps in time $k-1$ and k and two adjacent reference frames $\underline{\mathbf{m}}_1$, $\underline{\mathbf{m}}_2$, this computation results in:

$$\psi = \angle \left(\mathbf{P}(\underline{\mathbf{t}}_1^k, \underline{\mathbf{t}}_2^k)\, \mathbf{P}(\underline{\mathbf{t}}_1^{k-1}, \underline{\mathbf{t}}_1^k)\, \underline{\mathbf{b}}_1^{k-1}, \underline{\mathbf{b}}_2^k, \right). \tag{3.14}$$

With a twist stiffness β, the final twist energy between the edges results in

$$V_t = \frac{1}{2\,l_0} \beta (\theta_2 - \theta_1 + \psi)^2. \tag{3.15}$$

Note that twist forces acting on the centerline are non-zero, due to the influence of the centerline on the reference twist ψ. Similarly, bending forces acting on roll angles are non-zero, due to the influence of these angles on the orientation of material normals and binormals.

Material parameterization

The elliptical cross-section of the rod is discretized at edges with normal and binormal radii r_n and r_b. The area of the resulting cross-section is $A = \pi\, r_n\, r_b$. Knowing the Young modulus E and shear modulus G of the material, the stretch, bending, and twist stiffness can be computed, respectively, as [46]:

$$k_s = E\,A, \quad \mathbf{B} = \frac{E\,A}{4} \begin{pmatrix} r_n^2 & 0 \\ 0 & r_b^2 \end{pmatrix}, \quad \beta = \frac{G\,A(r_n^2 + r_b^2)}{4}. \tag{3.16}$$

The mass of the edges is lumped at nodes $\{\mathbf{c}_i\}$. For a material with mass density ρ, the lumped mass on a node with incident edge volumes V_1 and V_2 is $m = \rho(V_1 + V_2)/2$. Additionally, the model also considers the cross-sectional inertia of each twist angle $\{\theta_i\}$, which is $I = \rho V r_n r_b / 2$. Alternatively, $\{\theta_i\}$ variables might be assigned with zero inertia to ensure the material frames stay always in quasistatic equilibrium.

Simulation scheme

Independently of the nature of the simulation problem –whether we are solving dynamic deformation or static equilibrium–, generally each solver iteration provides a new guess for $\{\mathbf{c}_i\}$ and $\{\theta_i\}$. Once the new centerline is computed, we recompute the reference frame on every edge through parallel transport over time, and apply the roll angles to compute the new material frames. As noted before, the reference frame accumulates twist that must be updated after each iteration. For more details on the dynamic simulation algorithm, we refer the readers to the original work [173].

3.2.2 Rod network model

We propose an extension to discrete elastic rods model [21, 173] to handle connections between multiple rods. With the goal of avoiding the use of constraints, we devise an elastic energy that captures each rod deformation w.r.t. a rigid-body representation of the connection. Depending on how we keep track of connection kinematics, this framework leads to two alternative formulations:

- **A co-rotational approach**, that we defined in [22], where the orientation of each connection is estimated kinematically from incident rods. We formulate their implicit relation and show how the elastic coupling energy must be derived to correctly transfer bending and twist forces between connected rods.
- **An explicit approach**, later used in [50, 23], where each connection keeps track of its own orientation with 3 additional DoFs in the form of Euler angles.

In the following sections, we will first describe the co-rotational model, which benefits from a reduced number of DoF. Then, we will briefly explain the changes that are necessary to implement the explicit approach.

Coupling energy

For each rod incident in a given connection, we seek to define an energy that captures its deformation relative to all other incident rods. We follow a co-rotational approach, measuring deformation potentials w.r.t. rigidly transformed connections. Specifically, for each rod incident in a connection, we condense the effect of the remaining rods into a single *connection edge* whose orientation is determined kinematically using best-fit rigid transformations. Another possibility would be to define pairwise energies for the segments incident in a connection, as done in the *Cosserat nets* framework [47], but we found it more difficult to weight pair energy contributions to ensure a correct integration volume.

Let us consider a set of rod edges incident in a connection, with rest-shape material frames $\{\bar{\mathbf{m}}_i\}$. In the deformed setting, we estimate a rotation $\mathbf{R}$ of the rest-shape connection that best matches the deformed connection. Then, for each incident edge, we define a connection edge with material and reference frame $\mathbf{m}_{c,i} = \mathbf{R}\,\bar{\mathbf{m}}_i$, i.e., the rigidly rotated rest-shape material frame (Fig. 3.2). Note that, as material frames are already determined by the rotation of the connection, there is not need for a curve-angle discretization –i.e. reference and material frames are equal and hence roll angles at connection edges are trivially $\theta_{c,i} = 0$. However, the reference frame twist $\psi_{c,i}$ between each incident rod edge and its corresponding connection edge must be considered as for individual rods.

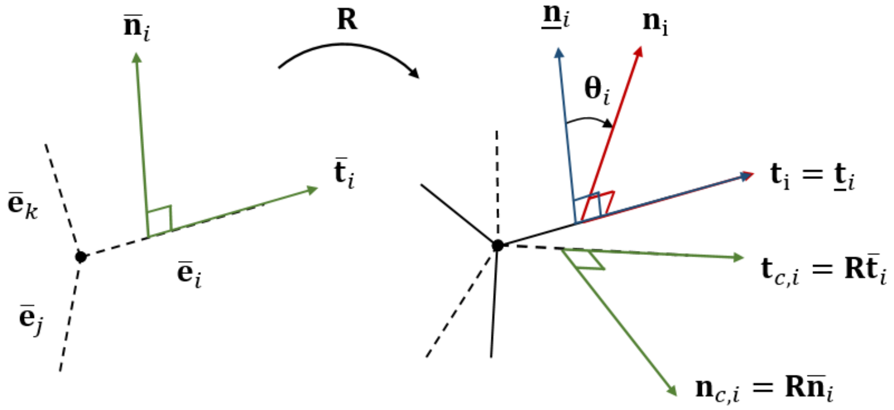

$\bar{\mathbf{n}}_i$
$\mathbf{R}$
$\underline{\mathbf{n}}_i$
$\mathbf{n}_\mathrm{i}$
$\boldsymbol{\theta}_i$
$\bar{\mathbf{t}}_i$
$\bar{\mathbf{e}}_k$
$\mathbf{t}_\mathrm{i} = \underline{\mathbf{t}}_i$
$\bar{\mathbf{e}}_i$
$\mathbf{t}_{c,i} = \mathbf{R}\bar{\mathbf{t}}_i$
$\bar{\mathbf{e}}_j$
$\mathbf{n}_{c,i} = \mathbf{R}\bar{\mathbf{n}}_i$

Figure 3.2: At any moment, the rigid state of the connection is determined by a rotation matrix $\mathbf{R}$ which is computed using shape matching. For each incident edge reference frame $\underline{\mathbf{m}}_i$ (blue) and corresponding material frame $\mathbf{m}_i$ (red), the rotation defines a connection edge with material frame $\mathbf{m}_{c,i} = \mathbf{R}\,\bar{\mathbf{m}}_i$ (green).

Given material and reference frames of incident edges and their connection edges, we can compute bending and twist elastic energies at the connections based on the deviations between incident edges and their corresponding connection edges. For doing so, we use the very same formula in Eqs. (3.13)-(3.15) with a slight difference: for each incident edge we use as integration domain half the rest length of the edge. Consequently the total potential elastic energy of the connection results:

$$V_c = \frac{1}{2}\sum_{i=0}^{k} V_b(\mathbf{m}_i, \mathbf{R}\,\bar{\mathbf{m}}_i) + V_t(\mathbf{m}_i, \mathbf{R}\,\bar{\mathbf{m}}_i), \tag{3.17}$$

where k is the number of incident rods in the connection. Note that the total domain of integration considered is half the sum of rest lengths of all edges incident in the connection. The deformation of each incident rod in the connection is affected by its own material properties and the anisotropic edge radii. This allows geometrically complex connections with heterogeneously thick incident rods to deform realistically according to the material, even for coarse discretizations (see Fig. 3.3 and Fig. 3.4).

Rotation estimation

In practice, we estimate the rotation $\mathbf{R}$ of a connection by minimizing the deformation of material frames of all incident edges. To consider the effect of both the bending and twist deformation of incident rods on the connection rotation, we measure the deviation of the material tangent $\mathbf{t}$ and one of its orthonormal material axes $\mathbf{n}$. Similarly to the shape matching approach by Mueller et al. [175], we first estimate the linear transformation that minimizes the least-squares deformation energy:

$$\mathbf{A} = \arg\min \sum_{i=0}^{k} \mathrm{Tr}(\mathbf{B}_i)\|\mathbf{t}_i - \mathbf{A}\,\bar{\mathbf{t}}_i\|^2 + \beta_i\|\mathbf{n}_i - \mathbf{A}\,\bar{\mathbf{n}}_i\|^2, \tag{3.18}$$

and then extract the rotation through polar decomposition $\mathbf{A} = \mathbf{R}\,\mathbf{S}$. Weighting the deformations of tangents and their orthonormal axes with each edge's bending and twist stiffness respectively, we favor the alignment of thicker and stiffer edges. Note that, for simplicity, both bending directions are considered together for the alignment of the tangent, and hence weights are not affected by the anisotropy of the rod and are just proportional to its thickness, Young modulus and shear modulus. A more precise alternative might have been considered by taking into account also the binormal of the incident edge albeit at the expense of a much more complex derivation of coupling forces.

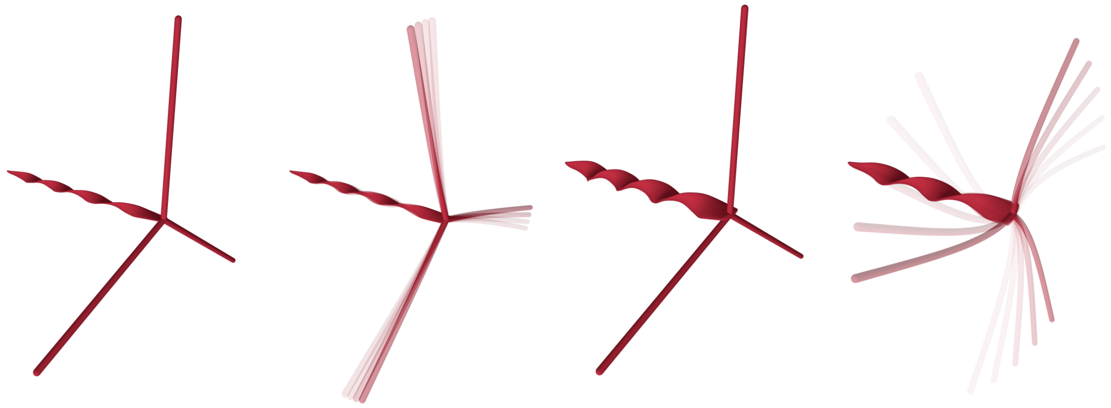


Figure 3.3: This example depicts force transfer at rod connections with our connection model. The axis of a windmill is twisted, and upon release its twist transfers into bending of the blades. A thinner axis (left) produces less momentum than a thicker one (right), as one would expect. The centerline of the axis was kept fixed.

Coupling Forces

To compute the bending and twist forces at connections, the respective energy terms need to be differentiated w.r.t. the DoFs of all incident rod edges. The difficulty stems from the fact that the material frames are not explicit DoFs, they are kinematically defined by the best-fit rigid transformation as explained above.

As material frames depend through the rotation of the connection on all DoFs of incident edges, the bending and twist energies of an edge pair formed by an incident edge and its connection edge need to be differentiated w.r.t. centerline coordinates and roll angles of all the incident edges of a connection, producing non-zero bending and twist forces between all pairs of incident edges. This is actually the expected behavior, because bending of one incident edge may produce twist on other incident edges and viceversa. Fig. 3.3 shows how our model succeeds to represent the coupling between rods incident in a connection.

Even though the number of force terms and required derivative calculations grows w.r.t. the regular discrete elastic rods method, these derivatives are mostly built from the same basic blocks. The only exception is the need to compute partial derivatives of rotated rest-shape material frames, $\mathbf{m}_c = \mathbf{R}\,\bar{\mathbf{m}}$, w.r.t. centerline coordinates and roll angles of all incident edges in the connection, as discussed in the paragraph above. By the chain rule, the differentials of rotated rest-shape material frames can be generally computed as:

$$\mathrm{d}(\mathbf{R}\,\bar{\mathbf{m}}) = \frac{\partial \mathbf{R}}{\partial \mathbf{A}} \sum_{i=0}^{k} \left(\frac{\partial \mathbf{A}}{\partial \mathbf{t}_i} \mathrm{d}\mathbf{t}_i + \frac{\partial \mathbf{A}}{\partial \mathbf{n}_i} \mathrm{d}\mathbf{n}_i \right) \bar{\mathbf{m}}_i. \tag{3.19}$$

$\frac{\partial \mathbf{R}}{\partial \mathbf{A}}$ is the derivative of polar decomposition which can be found in the work by Barbic and Zhao [78], and $\frac{\partial \mathbf{A}}{\partial \mathbf{t}_i}$ and $\frac{\partial \mathbf{A}}{\partial \mathbf{n}_i}$ are constant and easily computed from the linear expression that defines $\mathbf{A}$ [175]. Derivatives $\mathrm{d}\mathbf{t}_i$ and $\mathrm{d}\mathbf{n}_i$ w.r.t. incident edges centerline and roll angles can be easily derived from the discrete rod kinematics formulation introduced in the previous section 3.2.1.

Finally, static equilibrium solving requires the computation of energy Hessians, i.e., force Jacobians. Other than derivatives already present in the original discrete elastic rod formulation, our rod networks require the computation of the second derivatives of polar decomposition. Barbic and Zhao [78] also derived the scalar second derivative of polar decomposition, but we require a matrix of second derivatives with mixed terms and follow a very similar derivation.

We denote with $\mathrm{sk}(\boldsymbol{\omega})$ the 3×3 skew-symmetric matrix of a vector $\boldsymbol{\omega} \in \mathcal{R}^3$, i.e., $\mathrm{sk}(\boldsymbol{\omega})\,\mathbf{x} = \boldsymbol{\omega} \times \mathbf{x}, \forall \mathbf{x} \in \mathcal{R}^3$. Similarly, we denote with $\mathrm{sk}^{-1}(\mathbf{A})$ the unique skew-vector $\boldsymbol{\omega} \in \mathcal{R}^3$, such that $\mathrm{sk}(\boldsymbol{\omega}) = \frac{1}{2}(\mathbf{A} - \mathbf{A}^T)$.

Given a polar decomposition $\mathbf{A} = \mathbf{R}\,\mathbf{S}$, its first derivative is:

$$\begin{aligned}
&\frac{\partial \mathbf{R}}{\partial u} = \mathrm{sk}(\boldsymbol{\omega}(u))\,\mathbf{R}, \qquad \frac{\partial \mathbf{S}}{\partial u} = \mathbf{R}^T\left(\frac{\partial \mathbf{A}}{\partial u} - \frac{\partial \mathbf{R}}{\partial u}\mathbf{S}\right), \\
&\text{with } \boldsymbol{\omega}(u) = 2\,\mathbf{G}^{-1}\,\mathrm{sk}^{-1}\left(\mathbf{R}^T\frac{\partial \mathbf{A}}{\partial u}\right), \\
&\text{and } \mathbf{G} = (\mathrm{Tr}(\mathbf{S})\mathbf{I} - \mathbf{S})\mathbf{R}^T.
\end{aligned} \tag{3.20}$$

The mixed second derivative of the rotation matrix is:

$$\begin{aligned}
&\frac{\partial^2 \mathbf{R}}{\partial u \partial v} = \mathrm{sk}\left(\frac{\partial \boldsymbol{\omega}(u)}{\partial v}\right)\mathbf{R} + \mathrm{sk}(\boldsymbol{\omega}(u))\,\mathrm{sk}(\boldsymbol{\omega}(v))\,\mathbf{R}, \\
&\text{with } \frac{\partial \boldsymbol{\omega}(u)}{\partial v} = 2\,\mathbf{G}^{-1}\,\mathrm{sk}^{-1}\left(\mathbf{R}^T\left(\frac{\partial^2 \mathbf{A}}{\partial u \partial v} - \mathrm{sk}(\boldsymbol{\omega}(v))\frac{\partial \mathbf{A}}{\partial u}\right)\right) \\
&\quad - \mathbf{G}^{-1}\left(\mathrm{Tr}\left(\frac{\partial \mathbf{S}}{\partial v}\right)\mathbf{I} - \frac{\partial \mathbf{S}}{\partial v}\right)\mathbf{R}^T\boldsymbol{\omega}(u) + \mathrm{sk}(\boldsymbol{\omega}(v))\,\boldsymbol{\omega}(u).
\end{aligned} \tag{3.21}$$

Fig. 3.4 shows the application of our connection model to the dynamic simulation of a sphere-shaped rod mesh. The sphere moves down straight during compression, but then it twists up due to the mesh topology and the highly anisotropic bending behavior which favors tangent plane deformations.

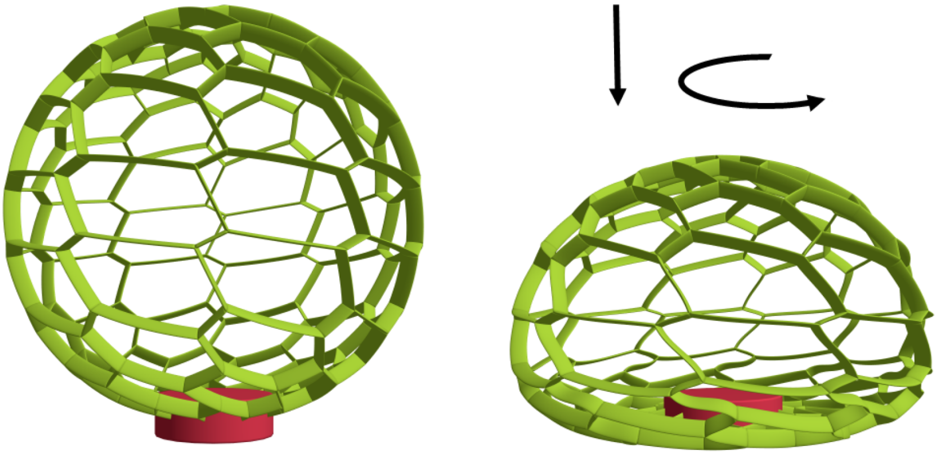


Figure 3.4: Dynamics of a sphere with complex interplay of compression and twist. The mesh topology and the highly anisotropic bending behavior favor tangent plane over out-of-plane deformations.

Explicit formulation

A formulation extending our model was proposed by Zehnder et al [50]. In this approach, instead of implicitly inferring connection states using best-fit transformations to incident rods, they explicitly modeled connections as mass-less rigid bodies with rotational DoFs. More precisely, the material frame at connection edges is computed as follows:

$$\mathbf{m}_c = \mathbf{R}_e(e_1, e_2, e_3)\, \mathbf{R}_c\, \bar{\mathbf{m}}. \tag{3.22}$$

Here, $\mathbf{e} = (e_1, e_2, e_3)$ are Euler angles which are used as rotational DoFs and $\mathbf{R}_e$ is the corresponding transformation. To avoid gimbal lock, after each step, the rotation from the Euler angles is transferred into a cumulative rotation $\mathbf{R}_c$ which is treated as a constant value. The total orientation is then the composition of the Euler angles and the cumulative rotation.

Despite the need of additional DoFs, derivatives of rigid-body rotations are computationally less expensive than calculating the expressions in Eqs. (3.19)-(3.21), and also more robust to extremely deformed configurations. In general, this leads to better conditioned optimization problems. We adopt this alternative formulation in our own work in chapter 6, because the designed rod meshes are sparsely connected, and so the computational overload due to the additional DoFs is negligible.

3.3 Thin-shell model

Thin shells are thin flexible structures with a high ratio of width to thickness (e.g. >100). Thin-shell models are usually governed by *membrane* (stretch) and *flexural* (bending) energies formulated based on nonlinear continuum mechanics and discretized using FEM. In the following sections, we will explain in detail the two mechanical formulations that we have used to model the membrane and flexural components of the energy:

- **Stretch** (section 3.3.1). We use a *constant-strain triangle* FEM discretization with a *St. Venant-Kirchhoff* (StVK) constitutive model. Although the material that we employ for fabrication is almost isotropic, our implementation considers also orthotropic effects [64] in order to accommodate other types of fabrics, if desired.

- **Bending** (section 3.3.2). We adopt the *discrete shells* model by Grinspun et al [63]. In our setting of cloth simulation, internal forces are strongly dominated by membrane stress and flexural stress is almost negligible. However, to avoid numerical problems when the fabric sags, we add weak bending forces.

3.3.1 Orthotropic membrane

A material is considered orthotropic if there exist three planes –orthogonal to each other– by reflections with respect to which material properties remain invariant [176]. The axes normal to these planes are called principal material directions. This is the case for many fabric materials, as principal axes are usually in correspondence with warp and weft directions in woven fabrics, and with course and wale directions in knitted fabrics.

Smooth formulation

Elastic surfaces resist stretching (local change in area) and shearing (local change in length but not area). In the classical thin plates and shells theory, see e.g. [177], the orthotropic material behavior is described by the generalized Hooke's law assuming a linear relationship between the strain and the stress tensor.

We consider an elastic surface a 2-manifold embedded in the 3D space with planar undeformed configuration. Extending the notation introduced in section 3.1, the deformation of an object in the continuum is often characterized by a time dependent deformation map $\phi : \mathcal{R}^2 \rightarrow \mathcal{R}^3$, from the material configuration $\bar{\mathbf{x}} \in \mathcal{R}^2$ to the deformed configuration $\mathbf{x} \in \mathcal{R}^2$.

Continuum elasticity is generally built upon the *deformation gradient* $\mathbf{F} = \nabla_{\bar{\mathbf{x}}}\phi$, i.e. the derivative of the deformation map w.r.t. material coordinates. In the case of the considered elastic surface, this is a 3×2 tensor. Based on the deformation gradient, rotation invariant measures are generally calculated to quantify the deformation, since pure rotation should not induce any stress in a deformable body. The most popular choice is the Green deformation tensor, which is defined as $\boldsymbol{\epsilon}_{2\times 2} = (\mathbf{F}^T\mathbf{F} - \mathbf{I})$. The material constitutive model then defines the relation between such strain and the Cauchy stress $\boldsymbol{\sigma}_{2\times 2}$, which provides the elastic forces per length in a unit material direction $\mathbf{n}$ as $\mathbf{f_n} = \boldsymbol{\sigma}\mathbf{n}$. The St. Venant-Kirchhoff model states that the strain and the stress are linearly related through a fourth order compliance tensor $\mathbf{C}$ like $\boldsymbol{\epsilon} = \mathbf{C}\boldsymbol{\sigma}$. This relation can be written in matrix form referred to the principal material axes:

$$\begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{12} \end{bmatrix} = \begin{bmatrix} \frac{1}{E_1} & -\frac{\mu_{21}}{E_2} & 0 \\ -\frac{\mu_{12}}{E_1} & \frac{1}{E_1} & 0 \\ 0 & 0 & \frac{1}{2G} \end{bmatrix} \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix}, \tag{3.23}$$

where we have unrolled the 2×2 symmetric matrices into 3-vectors. Components $11, 22$ are called normal components whereas component 12 is referred to as shear component. As the resulting matrix is block diagonal, it is easily invertible and the corresponding elasticity tensor (the so-called stiffness matrix) $\boldsymbol{\sigma} = \mathbf{K}\boldsymbol{\epsilon}$ is:

$$\mathbf{K} = \gamma \begin{bmatrix} E_1 & E_1\mu_{21} & 0 \\ E_2\mu_{12} & E_2 & 0 \\ 0 & 0 & \gamma^{-1}2G \end{bmatrix} \tag{3.24}$$

with $\gamma = (1 - \mu_{12}\mu_{21})^{-1}$. Unlike isotropic materials, which are parameterized with a single Young modulus E and a Poisson's ratio μ per dimension, orthotropic membranes require two Young modulus E_i, two Poisson's ratios μ_{ij} and one shear modulus G. Moreover, under isotropy assumptions, shear modulus can be derived from Young modulus and Poisson's ratio, but when considering orthotropic materials, that condition does not hold. As not all parameter settings lead to stable simulations, we suggest following the approach by Li and Barbic [64] to ease material characterization. In this work, the authors take some assumptions and derive a new parameterization which is reduced to two Young moduli and a single Poisson's ratio. More, precisely, the following conditions must hold:

$$E_1 > 0, \quad E_2 > 0, \quad \mu_{12}\mu_{21} > 1, \quad \frac{\mu_{12}}{E_1} = \frac{\mu_{21}}{E_2}. \tag{3.25}$$

In general, these conditions derive from enforcing the symmetry and positive-definiteness of the stiffness matrix. Under such conditions, parameters are transformed and the final formulation of the strain-stress relation is:

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} = \gamma \begin{bmatrix} E_1 & \sqrt{E_1E_2}\mu & 0 \\ \sqrt{E_1E_2}\mu & E_2 & 0 \\ 0 & 0 & \gamma^{-1}2G \end{bmatrix} \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{12} \end{bmatrix}, \tag{3.26}$$

$$\gamma = \frac{1}{(1+\mu)(1-\mu)}, \quad G = \frac{\sqrt{E_1E_2}}{2(1+\mu)},$$

where the simplified parameters are $\mu \in (0, 0.5)$ and $E_1, E_2 > 0$. Given this strain-stress relation, the corresponding membrane stretch energy density is computed as $V_s = \frac{1}{2}\sigma : \epsilon$, where the operator : represents the scalar product between the two tensors (i.e., the coefficient sum of the coefficient-wise product):

$$V_s = \frac{1}{2}\sum_{i=1}^{2}\sum_{j=1}^{2}\epsilon_{ij}\sigma_{ij}. \tag{3.27}$$

Expanding this expression and grouping terms, the elastic potential is:

$$V_s = \left(\frac{1}{2}\gamma E_1\right)\epsilon_{11}^2 + \left(\frac{1}{2}\gamma E_2\right)\epsilon_{22}^2 + \left(\gamma\mu\sqrt{E_1E_2}\right)\epsilon_{11}\epsilon_{22} + (2G)\,\epsilon_{12}^2.$$

When considering $E_1 = E_2$, this equation reduces to the classic St.VK elastic energy.

Discrete formulation

We represent the membrane as a 2-manifold triangle mesh $\mathcal{M}$ and discretize the continuum mechanics using a constant-strain triangle FEM. Information of the membrane state is stored on a finite number of 3D points $\mathbf{x}_i \in \mathcal{R}^3$, with planar material configuration $\bar{\mathbf{x}}_i \in \mathcal{R}^2$. Finite elements specify a method for reconstructing a continuous deformation map $\hat{\phi} : \mathcal{R}^2 \rightarrow \mathcal{R}^3$ through interpolation from discrete values $\mathbf{x}_i = \phi(\bar{\mathbf{x}}_i)$. Let us consider a generic scheme in which nodal values of a given variable, e.g., position $\mathbf{x}$, are expressed throughout the element via polynomial interpolation as:

$$\mathbf{x}(\bar{\mathbf{x}}) = \sum_{i=1}^{n_v} \mathbf{x}_i \, N_i(\bar{\mathbf{x}}), \tag{3.28}$$

where n_v is the number of vertices in the element and each $\mathbf{N}_i(\bar{\mathbf{x}})$ (called shape function) is the interpolation weight associated with the $i - th$ node. Note that we explicitly show the dependence of the shape function on the material coordinate $\bar{\mathbf{x}}$ corresponding to the interpolated point $\mathbf{x}$. In matrix form, this results in $\mathbf{x}(\bar{\mathbf{x}}) = \mathbf{D}_w \, \mathbf{N}(\bar{\mathbf{x}})$, where all spatial positions of the element vertices have been assembled into a $3 \times n_v$ matrix $\mathbf{D}_w = [\mathbf{x}_1, \ldots, \mathbf{x}_{n_v}]$ and, similarly, shape function values have been assembled in a $n_v \times 1$ vector $\mathbf{N} = [N_1, \ldots, N_{n_v}]^T$. In this context, a discrete approximation of the deformation gradient can be computed as:

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \bar{\mathbf{x}}} = \mathbf{D}_w \, \frac{\partial \mathbf{N}(\bar{\mathbf{x}})}{\partial \bar{\mathbf{x}}} = \mathbf{D}_w \, \mathbf{B}(\bar{\mathbf{x}}). \tag{3.29}$$

Here, $\mathbf{B}$ is the $n_v \times 2$ matrix of partial derivatives of shape functions w.r.t. material coordinates. Finite elements method highly depends on the particular choice of the interpolation function. In the case of a triangular mesh, barycentric interpolation would be the natural election to extend the nodal deformations to the entire interior of the mesh. For this particular case, the matrix $\mathbf{B}$ does not depend on the material coordinates of the point, and hence the deformation gradient has a closed form and can be computed as follows:

$$\mathbf{F} = \mathbf{D}_w \, \mathbf{G} \, (\mathbf{D}_m \mathbf{G})^{-1}, \quad \mathbf{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix} \tag{3.30}$$

where all material positions of the element vertices have been assembled into a $2 \times n_v$

matrix $\mathbf{D}_m = [\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_{n_v}]$. The value of the matrix $\mathbf{G}$ is derived from the material derivative of shape functions corresponding to isoparametric elements [178]. We found this formulation to be the most generic and extensible to other interpolation schemes and discretization elements, but there are other equivalent derivations for the deformation gradient. For instance, we suggest the reader the course by Sifakis and Barbic [179] for further information.

With a discrete approximation of the deformation gradient, it is then straightforward to compute the elastic potential energy in (3.28) and derive the corresponding force and Jacobian expressions needed for implicit integration.

3.3.2 Discrete shells bending

For completeness, we present here an overview of the thin shells bending model that we use. Flexure models in shell mechanics are usually based on the difference of the second fundamental forms between the deformed and undeformed configurations. This tensorial expression measures deformation and is invariant to rigid-body transformations of the coordinate frame. Same as with the membrane energy, these treatment derives expressions over smooth manifolds and then discretizes to carry out the numerics. But because of their degeneracy in one dimension, the development of robust finite element methods for thin shell-equations is still challenging.

In contrast, the work we adopt, discrete shells by Grinspun et al. [63], directly defines a constitutive model based on discrete geometric operators applied over piecewise-linear surfaces. The resulting expressions are easy to implement and capture the deformation behavior of the cloth well enough to prevent wrinkling and improve the numerical conditioning of the simulation problem.

The discrete shells bending energy is based on the difference between the shape operator –which measures the local curvature at a point on a smooth surface– evaluated on the deformed and undeformed surfaces. Several discrete approximations to this magnitude can be found, e.g., in [180]. In discrete shells, the authors propose to discretize the shape operator at edges, measuring the dihedral angle in the deformed θ and undeformed $\hat{\theta}$ configurations. For a given edge $\mathbf{e}$, the dihedral angle is the angle between the vectors normal to its neighboring faces. Considering this definition, the discrete flexural energy is the summation over mesh edges of the expression:

$$V_b = \frac{||\bar{\mathbf{e}}||}{\bar{h}_e} (\theta_e - \bar{\theta}_e)^2, \tag{3.31}$$

where $\bar{h}_e$ is a third of the average of the heights of the two triangles incident to the edge. Implementation-wise, this discrete energy is straightforward to implement, but depending on the method used for the measurement of the dihedral angle, it might be necessary to include safeguards for numerical robustness.

3.4 Static equilibrium

This thesis deals with design problems for which the main performance criterion is the deformation behavior, more precisely, we consider the equilibrium of fabricated objects for any given boundary conditions and not how they dynamically behave. Technically, the static equilibrium is defined by the shape with vanishing net forces, that is $\mathbf{f}(\mathbf{q}) = \mathbf{0}$. As introduced in section 3.1, $\mathbf{q} \in \mathcal{R}^n$ is the deformed configuration of the object in some known generalized coordinates system. The resulting force $\mathbf{f} \in \mathcal{R}^n$ is generally a nonlinear vector function that may include internal forces $\mathbf{f}_i$ (e.g., elastic forces) and external forces $\mathbf{f}_e$ (e.g., gravity force). In general, forces depend on the deformed configuration $\mathbf{q}$, as well as on some other magnitudes (e.g., material parameters, gravity constant, etc.) that are kept constant during simulation.

Design methods rely on numerical simulation to predict the actual behavior of fabricated designs. For some specific boundary conditions, we seek to compute the corresponding deformed configuration $\mathbf{q}$ in the most accurate and efficient manner possible. In the following sections, we will overview a two approaches for the implementation of the static equilibrium solver that might be considered and discuss their practical applicability.

3.4.1 Root-finding solver

Using classical Newtonian mechanics, computing the static equilibrium means solving a system of nonlinear equations or, equivalently, finding the roots of the vector-valued function $\mathbf{f}$. The general approach is to use the Newton-Raphson method to calculate successively better approximations to the function roots. Roughly, given an initial estimate $\mathbf{x}_0$, successive solutions are computed:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{J}^{-1}(\mathbf{x}_k)\ \mathbf{f}(\mathbf{x}_k), \tag{3.32}$$

where $\mathbf{J} = \nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x}_k)$ is the Jacobian of the forces w.r.t. generalized coordinates. In practical terms, that means iteratively solving linear systems of the kind $\mathbf{J}_k\Delta\mathbf{x} = -\mathbf{f}_k$, where $\Delta\mathbf{x} = (\mathbf{x}_{k+1} - \mathbf{x}_k)$ and functions dependent on generalized positions at the $k - th$ iteration have been shortened to $\mathbf{f}_k$ and $\mathbf{J}_k$. In certain situations, the Newton-Raphson method is an extremely powerful technique and has local quadratic convergence.

Practical problems

However, there are practical considerations –related mainly with the properties of the Jacobian– that might lead to the failure of the Newton-Raphson method. In the following, we review the main practical issues that must be considered when solving static equilibria, specially with the root-finding algorithms:

- **Unstable equilibria**. An equilibrium is unstable if a small position perturbation will likely produce a large change in the energy of the system. Mechanical equilibria happen at the critical points of the potential energy functions governing the system. These can be either local minima, local maxima or saddle points. Both maximum and saddle points are representative of unstable equilibria. These situations are rather usual, specially in the case of mechanical models where highly anisotropic behaviors are present, and can easily produce *buckling effects*. For example, when a stiff elastic rod is subject to axial compressive stress, buckling may occur and produce a sudden sideways deflection. Although these are valid equilibrium configurations, the static solver should avoid moving towards them. This is justified because not only they are physically unstable and do not represent a statistically probable state for a fabricated object, but they are also numerically related with indefinite points.

- **Indefinite points** These correspond to points of the space for which the Jacobian is indefinite. Indefinite matrices are characterized by having both positive and negative eigenvalues. This limits the range of linear solvers that can be used to solve the linear system in (3.32), as most used direct and iterative methods require a positive semi-definite matrix to properly function, e.g., Cholesky and Conjugate Gradient. In addition, an indefinite point indicates the presence of a saddle point in the potential energy function, and so an unstable equilibrium.

- **Stationary points**. These are points of the generalized space at which the Jacobian is singular, and consequently the linear system in (3.32) cannot be solved, causing the algorithm to halt. Analytically, this implies that the null-space of the Jacobian matrix is non-empty, i.e., there exist directions in the space for which the value of the force function does not change. Therefore, the problem has infinite solutions. In mechanical simulation, this usually means that numerical constraints in the form additional boundary conditions must be imposed to avoid the singularity. For instance, as we have explained in the previous section 3.3.1, most elastic energy formulations are invariant to rigid-body transformations and hence translation or rotation steps must be prevented.

- **Ill-conditioning**. Even if the Jacobian matrix is strictly non-singular and positive definite, mechanical models can usually lead to ill-conditioned problems. If the Jacobian is nearly singular or has a large condition number –ratio between the maximum and mimimum eigenvalue– at some iteration, the linear system is not numerically well behaved for the computation of a solution. This might lead to bad estimates that may overshoot the step and diverge from the closest root. A recent work by Kannan et al. [181] has analyzed potential causes and solutions for ill-conditioned problems in structural mechanics.

Practical solutions

Here we identify two solutions that may overcome the problems mentioned above:

- **Regularization** The challenges due to indefinite problems are usually addressed through linear system regularization. The simplest solution is solving the linear system $(\mathbf{J}+\lambda\mathbf{I})\Delta\mathbf{q} = -\mathbf{f}$ with increasingly higher values for λ, until the resulting matrix is strictly positive definite. Nevertheless, there are other regularization schemes that may also help with ill-conditioning. For instance, the well-known *Thikonov regularization* formulates an ordinary least squares problem to solve the linear system and adds a regularization term in order to give preference to particular solutions with desirable properties. This leads to a generic linear system formulation $(\mathbf{J}^T\mathbf{J}+\mathbf{Q}^T\mathbf{Q})\Delta\mathbf{q} = -\mathbf{J}^T\mathbf{f}$, for some matrix $\mathbf{Q}$. Depending on the choice of $\mathbf{Q}$, Thikonov regularization may improve the conditioning of the problem, thus enabling a direct numerical solution. For a more thorough study on regularization, we refer the reader to [182].

- **Step selection** Even with the previous safeguard, overshoot and divergence from the roots might still happen due to the nonlinearity of the force function and poor initial estimates. To mitigate this problem, a step length selection procedure should be also used. A straightforward solution could be to consider a simple backtracking line-search in the $\Delta\mathbf{q}$ direction that ensures the reduction of a merit function ϕ. Most literature on nonlinear equation system solving suggests using the norm of the function, $\phi = \|\mathbf{f}\|$. More sophisticated line-search methods consider additional convergence conditions such as the *Armijo rule* (i.e., sufficient decrease in the merit function) and the *Wolfe condition* (i.e., sufficient decrease in the curvature) [183]. However, independently on the particular line-search scheme used, Newton-Raphson algorithm together with this merit function is equally attracted to local minima and saddle points, what negatively affects the overall convergence towards good solutions. In the next section, we will introduce a very similar formulation that originates from Lagrangian mechanics and improves the robustness of the solver.

3.4.2 Optimization solver

As we have seen, the set of solutions to the static equilibrium problem correspond to the critical points of the potential energy function V. Therefore, we can reformulate the problem as the minimization

$$\min_{\mathbf{q}} V(\mathbf{q}), \tag{3.33}$$

which finds solution at the points where the gradient vanishes $\nabla_{\mathbf{q}} V = -\mathbf{f} = \mathbf{0}$. Minimization of the potential energy imposes a bias towards stable equilibrium configurations that avoid the problematic saddle points mentioned in the previous section. Formulating the problem as a nonlinear optimization offers the possibility of using a much broader range of numerical methods. In practical terms, we solve the problem in (3.33) using *sequential quadratic programming* (SQP), for which the step computation is formally equivalent to the root-finding Newton-Raphson method described above, –we refer to Nocedal and Wright [183] for formal proof. The only difference between the two is in the merit function used for the step length selection procedure, which in this case is $\phi = V$, to guarantee that each step produces a valid estimate closer to a stable static equilibrium configuration.

Chapter 4

Inverse elastic shape design

Our tools feature inverse design operators that automatically determine attributes of the elastic structures such that their deformed shapes meet some user-specified goals. The associated numerical optimization problem has been repeatedly formulated throughout the extensive literature, having become a standard scenario. In this chapter, we present a formal characterization of such problem, describe some numerical solving methods, and analyze the challenges of their practical implementation. Our goal is to address some common issues across many design problems that fit within this framework and discuss alternative solution approaches.

- The first section 4.1 formalizes the optimization problem to establish a common notation and characterize this problem based on its theoretical solution space.
- The second section 4.2 analyzes the practical implications of solving this optimization, provides some guidelines on how to formulate design problems to favor finding better solutions, and introduce two solution methodologies.
- The third section 4.3 focuses on computational solutions based on the exploration of the constraint manifold, with an emphasis on the usefulness of sensitivity analysis for that purpose, and discuss current issues with commonly-used standard optimization methods.

4.1 Problem characterization

In this thesis, we study the following generic problem: given a parameterized flexible object, find the design –i.e., parameter values– that optimize some metric dependent on the static equilibrium deformation of the object, for some boundary conditions.

In this section, we extend the generic notation introduced in the last chapter 3 in the context of mechanical simulation and formally characterize the problem to serve as a baseline for further analysis. We then study the theoretical solution space to better understand the underlying mathematical problem.

4.1.1 Problem formulation

We consider a generic computational model $\mathcal{M}(O)$ that describes the mechanical behavior of a deformable object $O \subset \mathcal{R}^3$ through a conservative energy potential $V(\mathbf{q}, \mathbf{p}) \in \mathcal{R}$. Here, $\mathbf{q} \in \mathcal{R}^n$ concatenates all the independent variables that discretize the deformed shape of the object in some known generalized coordinates system, and $\mathbf{p} \in \mathcal{R}^m$ is an arbitrary set of parameters that determines the design of the object for a particular problem. From here onward, we will refer as *deformed space* to the space $Q \equiv \mathcal{R}^n$ spanned by $\mathbf{q}$, *design space* to the space $P \equiv \mathcal{R}^m$ spanned by $\mathbf{p}$, and *total space* to the union of both $R \equiv P \cup Q \equiv \mathcal{R}^o$ spanned by $\mathbf{r} = (\mathbf{q}, \mathbf{p})$, with $o = n + m$. This parameter set might contain variables modeling discrete geometry (e.g., the rest-shape) or material properties (e.g., Young modulus), among others. We are omitting here other quantities that might affect the value of the energy potential (e.g., gravity constant), but whose value is not considered a design parameter.

Altogether, we then define the generic shape optimization problem as follows:

$$\begin{aligned} \min_{\mathbf{q}_i, \mathbf{p}} \ & \sum_{i=1}^{k_p} h(\mathbf{q}_i) \\ \text{s.t.} \ & \mathbf{f}_{\mathbf{q},i}(\mathbf{q}_i, \mathbf{p}) = \mathbf{0} \\ & \mathbf{p}_m \leq \mathbf{p} \leq \mathbf{p}_M. \end{aligned} \tag{4.1}$$

This formulation minimizes an objective function h dependent on a number of k_p deformed poses $\mathbf{q}_i$, subject to the static equilibrium constraint $\mathbf{f}_{\mathbf{q},i} = \mathbf{0}$ and box constraints on design parameters $\mathbf{p}_m \leq \mathbf{p} \leq \mathbf{p}_M$. While the deformed configuration $\mathbf{q}_i$ is generally unbounded, design parameters $\mathbf{p}$ are often restricted to valid ranges. For instance, materials are characterized by Young's modulus $E \in (0, \infty)$ and Poisson's ratio $\mu \in (0, 0.5)$. For this reason, lower $\mathbf{p}_m$ and upper $\mathbf{p}_M$ limits are considered. Note that the equilibrium constraint corresponding to each pose $\mathbf{f}_{\mathbf{q},i}$ considers its own boundary conditions. This generic formulation allows, for instance, to define the desired range of deformations of a fabricated object based on its deformed configurations corresponding to multiple scenarios, as it will be demonstrated in chapter 5. In the following sections, we will further explain both the static equilibrium constraint and the objective function h.

We note that the formulation of inverse elastic design problems might be more general, containing also other types of constraints. However, for the purpose of our analysis, it suffices to consider the problem (4.1) above.

The static equilibrium

We deal with design problems for which the main performance criterion is the deformation behavior. More precisely, we consider the equilibrium of fabricated objects for any given boundary conditions, i.e., the shape with vanishing net forces. The defined energy potential yields the expression of the system forces as

$$\begin{aligned}\mathbf{f_q}(\mathbf{q},\mathbf{p}) &= -\frac{\partial V}{\partial \mathbf{q}} \in \mathcal{R}^n, \\ \mathbf{f_p}(\mathbf{q},\mathbf{p}) &= -\frac{\partial V}{\partial \mathbf{p}} \in \mathcal{R}^m.\end{aligned} \tag{4.2}$$

Note that $\mathbf{f_q}$ are the net forces acting on the DoFs corresponding to the deformed shape. The analogous magnitude at design parameters $\mathbf{f_p}$ is not required at this point, but it will be of use in later sections. The partial derivative of both force expressions will be also needed and we will refer to them using the following notation:

$$\mathbf{J}(\mathbf{q},\mathbf{p}) = \begin{bmatrix} \mathbf{J_{qq}} & \mathbf{J_{qp}} \\ \mathbf{J_{pq}} & \mathbf{J_{pp}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{f_q}}{\partial \mathbf{q}} & \frac{\partial \mathbf{f_q}}{\partial \mathbf{p}} \\ \frac{\partial \mathbf{f_p}}{\partial \mathbf{q}} & \frac{\partial \mathbf{f_p}}{\partial \mathbf{p}} \end{bmatrix}. \tag{4.3}$$

The expression of static equilibrium $\mathbf{f_q}(\mathbf{q}_j,\mathbf{p}) = \mathbf{0}$ imposes numerical constraints on the total space R in the form of an underdetermined system of n –generally nonlinear– equations and o variables. The set of all possible configurations $(\mathbf{q},\mathbf{p})$ such that these conditions hold defines an m-manifold embedded in the total space R, which we call *equilibrium manifold*, $\mathcal{E}$. If the potential energy is conservative and bounded from below for a specific parameter vector $\mathbf{p}$, there is at least one equilibrium configuration that we denote:

$$\hat{\mathbf{q}} = \{\mathbf{q} \,|\, \mathbf{f_q}(\mathbf{q},\mathbf{p}) = 0\}, \tag{4.4}$$

located at an extremal point of the potential energy function. Recall from section 3.4 that both maxima and saddle points correspond to unstable static equilibrium states and are not desirable configurations. Additionally, the elastic potential is generally invariant to rigid-body transformations. Therefore, from this point onward, we assume that Q has an empty null-space, i.e., translation and rotation displacements have been already constrained through boundary conditions. In any case, in section 3.4, we described a numerical solver that can be used to compute such configurations robustly, navigating only the deformed space Q.

The objective function

In this kind of problems, the quality of a given solution is usually measured in terms of a number of k_s shape descriptors $\mathbf{\Phi}_i \in \mathcal{R}^{s_i}$ of an arbitrary dimensionality s_i. We call the space spanned by the combination of all possible $\mathbf{\Phi}_k$ vectors $S \equiv S_1 \cup \ldots \cup S_k$ the *shape space*.

The simplest setting would be the case where we seek design parameters such that the deformed shape of the object $\mathbf{q}$ exactly matches a given target configuration $\mathbf{q}^t$. Then the identity function is the unique shape descriptor $\Phi(\mathbf{q}) = \mathbf{q}$, in which case $s = n$ and the shape space equals the deformed space $S \equiv Q$. However, more often objectives focus on a subset of the deformation variables, $\Phi(\mathbf{q}) = \mathbf{T}\,\mathbf{q}$, with $\mathbf{T}_{s\times n}$ being a selection matrix. That is the case of the works that will be described in chapters 5 and 6, for instance, where we are interested only in the positions of the centerline nodes of a rod mesh, while the deformed shape is also determined by the roll angles of the material frame. In general, any combination of k shape descriptors could potentially be used in the objective function, including more sophisticated metrics like area, volume, smoothness, stability or buoyancy, among others. For each of these shape descriptors, one must provide a corresponding target value $\mathbf{\Phi}^t$, the distance to which we aim to minimize. This defines a generic objective function as the squared difference between shape descriptors and target values

$$h = \frac{1}{2}\sum_{j=1}^{k_s} w_j \,\|(\mathbf{\Phi}_j(\mathbf{q}_i) - \mathbf{\Phi}^t_{ij})^T\|^2, \tag{4.5}$$

where w_j are descriptor weights.

Basic problem formulation

For notation simplicity and without loss of generality, further analysis will consider only one pose $k_p = 1$ and one shape descriptor $k_s = 1$, which will be the identity $\mathbf{\Phi}(\mathbf{q}) = \mathbf{q}$. This way, $n = s$ and the shape space is equivalent to the deformed space $S \equiv Q$. With that in mind, we finally define the shape optimization problem as follows:

$$\begin{aligned} \min_{\mathbf{q},\mathbf{p}} \; & \frac{1}{2}\|\mathbf{q} - \mathbf{q}^t\|^2 \\ \text{s.t.} \; & \mathbf{f}_{\mathbf{q}}(\mathbf{q},\mathbf{p}) = \mathbf{0} \\ & \mathbf{p}_m \leq \mathbf{p} \leq \mathbf{p}_M. \end{aligned} \tag{4.6}$$

In the following section we will analyze the theoretical solution space of this optimization problem and characterize inverse elastic design problems accordingly.

4.1.2 The solution space

In general, equation (4.6) defines a nonlinear least squares problem that can be solved using a standard constrained nonlinear optimization method. In this thesis, we focus on a solution scheme based on navigating the equilibrium constraint manifold. In this section, we study the solution space of the problem in order to better understand the geometry of such constraint manifold and how it is related to the objective function.

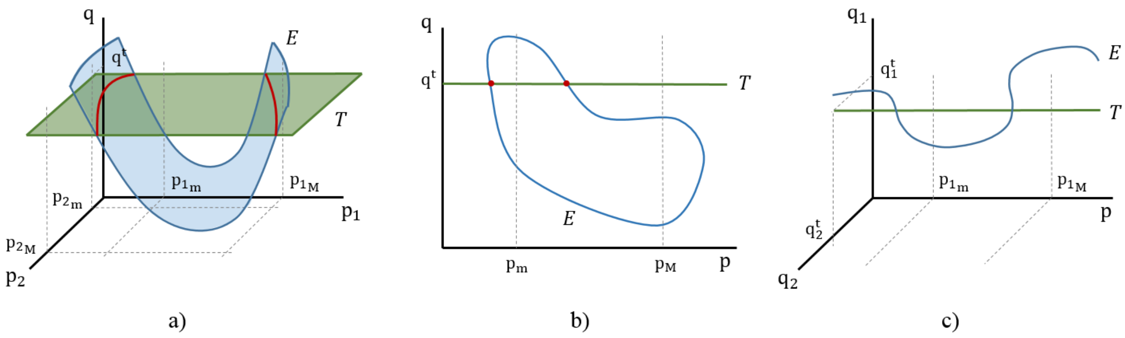


Figure 4.1: This figure represents the three possible situations that might happen depending on the dimension of design P and shape S spaces. The set of perfect solutions (red) is defined by the intersection between the target (green) and equilibrium (blue) manifolds. Note that manifolds do not intersect in the c) case.

The objective function in the problem (4.6) trivially finds a minimum at $\mathbf{q} = \mathbf{q}^t$. This condition defines a m-manifold embedded in the total space R, where the objective function vanishes, which we call *target manifold*, $\mathcal{T}$. Note that, as the objective function does not depend on the design parameters, this target manifold is constant in the P subspace. The intersection between equilibrium $\mathcal{E}$ and target $\mathcal{T}$ manifolds defines the set of perfect solutions with $h = 0$. In general, the maximum dimension of the intersection between two manifolds $\mathcal{A}$, $\mathcal{B}$ embedded in a space R is a non-negative value formulated as $\mathrm{Dim}(\mathcal{A} \cap \mathcal{B}) = \mathrm{Dim}(\mathcal{A}) + \mathrm{Dim}(\mathcal{B}) - \mathrm{Dim}(R)$, which in the case of our problem results $\mathrm{Dim}(\mathcal{E} \cap \mathcal{T}) = \max(m - n, 0)$. Considering this fact, inverse design problems can be also characterized in terms of the dimensionality of the design P and shape S spaces. Fig. 4.1 shows a geometric representation of the three situations that might happen:

- **Case Dim**(P) $>$ **Dim**(S). There are potentially infinite perfect solutions. One instance of this problem appears in some interactive computational tools that feature inverse design capabilities, see e.g. [14, 26, 15, 23]. In these cases, the user sequentially applies geometric changes to the deformed configuration $\mathbf{q}$, e.g., pulling from a mesh vertex handler, to define a new target $\mathbf{q}^t = \mathbf{q}+\Delta\mathbf{q}^t$. The objective function is then formulated to match the incremental change locally, resulting in a low-dimensional shape space. To select among the infinite possible solutions, regularizing terms must be added to the objective function, e.g., to favor the closest solution to the starting point.

- **Case $\mathbf{Dim}(P) = \mathbf{Dim}(S)$.** There is at most a finite number of perfect solutions and, under some assumptions on the normal and curvature of the equilibrium manifold, it can be shown that a solution exists. This situation corresponds, for example, to design problems where the rest configuration is optimized such that the deformed configuration matches a desired shape $\mathbf{p} = \bar{\mathbf{q}}$. In absence of external forces, this problem is trivial, i.e., $\mathbf{p} = \mathbf{q} = \mathbf{q}^t$, With external forces, the problem is not trivial and it has been often studied, e.g., [140, 139, 9].

- **Case $\mathbf{Dim}(P) < \mathbf{Dim}(S)$.** There is at most a finite number of perfect solutions, but the set is generally empty. This is the most common example in the design literature. In material characterization, for instance, only a few material parameters are considered as defining the design space, but the shape descriptor often considers the whole deformed configuration, e.g., [148, 68, 184]. In more general fabrication-oriented design works, the complexity of solving (4.6) rapidly grows with the size of the design space, and dense parameterizations are more likely to produce noisy results. Hence, there is a general interest in using low-dimensional design spaces.

We may assume that, in most cases, there will be no intersection between the two manifolds. Note that, even when $m >> n$, the target configuration might not be reachable because we are restricted to the feasible range of design parameters $[\mathbf{p}_m, \mathbf{p}_M]$. Even if the target configuration is realizable, the equilibrium manifold $\mathcal{E}$ is generally nonlinear and we can only make a guess about the intersection set based on local linear approximations. In practice, we will look for a point in the equilibrium manifold that locally minimizes the distance function h. In section 4.2, we will explore different alternatives for finding this solution.

4.2 Solution methodologies

The optimization problem defined in (4.6) can be solved using standard optimization methods for general nonlinear least-squares problems. However, we have seen that the equilibrium manifold that defines the solution space is often nonlinear and full of perils. Additionally, the target shape can be arbitrarily far from any static equilibrium configuration. In general, this makes the design problem very difficult and produces misleading results when computing the numerical approximations needed to solve the problem.

During the development of this thesis, we have dealt with two main aspects that largely affect the complexity of the design problem and identified simple approaches that facilitate its solution: i) *the selection of the design space*, and ii) *the definition of the target shape*. In the next two sections 4.2.1 and 4.2.2, we will describe each one of them. Then, we will introduce the two main approaches that we have considered for solving the optimization problem: constrained optimization, in section 4.2.3, and constraint manifold exploration, in section 4.2.4.

4.2.1 Design space selection

There are some aspects of the optimization problem that benefit from an adequate selection of the design parameters. In general, we aim for formulations that have two complementary attributes:

1. **Naturally smooth solutions**. Fabrication-oriented design applications generally benefit from producing smooth shapes and designs, not only for aesthetic reasons. The presence of high frequency noise in the shape or the material distribution of an object often reduces the accuracy of the discretizations used to approximate the continuum, produces numerical instabilities in the computation of the gradient and the Hessian, makes the fabricated object more prone to malfunction or fracture, and hinder the manufacturing process.

2. **Compact parameter sets**. In general, the complexity of an inverse elastic design problem rapidly grows with the size of the design space and hence there is an interest in using compact parameterizations. Although this actually involves shrinking the solution space and making it more difficult to find equilibrium configurations close to the target shape, it is less prone to overfitting, which is also associated with non-smooth solutions.

Consequently, we are interested in selecting a compact parameter set such that the part of the solution space that is not accessible corresponds mainly to non-smooth and overfitted solutions. One usual way of producing smooth solutions is to add regularization terms to the objective function, but these terms are usually not trivial to control and do not make the design space more compact. Instead of using regularization, we propose defining the design space as a numerical interpolation from a reduced set of parameter control points $\mathbf{p} = \mathbf{z}(\alpha)$, with $\alpha \in \mathcal{R}^a$ and $a << m$. The particular nature of the interpolation function $\mathbf{z}$ depends on the specific design problem. For instance, in chapter 5, we employ cubic Hermite interpolation to define the radii of a rod section along the centerline. Conversely, in chapter 6, we use Laplacian meshing to determine the rest configuration of the membrane of a tensile structure. In any case, it is convenient to choose interpolations where all nonlinearities can be precomputed and the final relationship is linear $\mathbf{p} = \mathbf{Z}_{m \times a}\alpha$. Hence, for any magnitude $\mathbf{w}$,

$$\frac{\partial \mathbf{w}}{\partial \alpha} = \frac{\partial \mathbf{w}}{\partial \mathbf{p}} \mathbf{Z}, \qquad \frac{\partial^2 \mathbf{w}}{\partial \alpha^2} = \mathbf{Z}^T \frac{\partial^2 \mathbf{w}}{\partial \mathbf{p}^2} \mathbf{Z}. \tag{4.7}$$

This relationship allows us to reformulate the optimization problem, without any loss of generality, in terms of the parameter interpolation control points α. To ease the notation though, we will continue using the basic formulation in (4.6).

4.2.2 Target shape definition

As we introduced in section 4.1.2, the specified target shape is most likely infeasible, independently of the parameterization of the problem. In fact, it might be arbitrarily far from any possible equilibrium configuration. Under such conditions, numerical optimization methods usually produce poor results. Introducing prior knowledge about the design problem in the definition of the target shape is generally necessary to obtain good solutions. We propose two approaches for this purpose: using physically-based targets or equilibrium manifold approximations.

- **Physically-based target** Often, shape optimization approaches try to match a target shape that has been defined by the user employing some geometry deformer. Alternatively, using physically-based deformers for this task highly increases the chances of producing target shapes that can be successfully matched within the considered design space. Even if the underlying mechanical model of the deformer does not accurately match the model that determines the equilibrium constraint, similar principles often apply in terms of volume conservation and deformation ratios. In chapter 5, we show how thin-shell and FEM computational models can be used to define a target shape that is successfully matched by a rod mesh model.

- **Manifold approximations** Direct freeform manipulation of the target is always possible (e.g., defining artistic shapes with an aesthetic intention), but it will most likely produce targets that are far from feasible. Instead, using an incremental target exploration ensures that the targets that are defined remain close to feasible. Starting from an initial feasible solution, one can compute local approximations of the equilibrium manifold and successively deform the target shape such that the approximate constraints hold. Once satisfied with the deformed shape, this can be used as the target configuration of the full nonlinear optimization problem. Although the resulting objective is probably still not feasible, the chances of the design optimization to find a valid approximation are much higher. In chapter 6, we present a two-step optimization that is based on this principle.

4.2.3 Constrained optimization solution

Many works in the literature solve the resulting constrained optimization problems using standard optimization methods, e.g., [8, 10, 31, 28]. Our constrained nonlinear least squares problem can be generally solved using a *sequential quadratic programming* (SQP) method. This common denomination includes a large variety of optimization algorithms based on iteratively approximating the nonlinear problem (4.6) with a simpler *quadratic programming* (QP) that is easier to solve. Let us assume for now that there are no box constraints, i.e., $\mathbf{p}_m = -\infty$ and $\mathbf{p} = \infty$. In such case, each SQP iteration implies:

1. A quadratic approximation of the objective function

$$\tilde{h}(\Delta\mathbf{r}; \mathbf{r}_0) = \mathbf{g}(\mathbf{r}_0)^T \, \Delta\mathbf{r} + \frac{1}{2} \, \Delta\mathbf{r}^T \, \mathbf{H}(\mathbf{r}_0) \, \Delta\mathbf{r}, \tag{4.8}$$

   where $\mathbf{g} = \nabla_{\mathbf{r}} \, h$ and $\mathbf{H} = \nabla_r^2 \, h$.

2. A linear approximation of the equilibrium constraint

$$\tilde{\mathbf{f}}_{\mathbf{q}}(\Delta\mathbf{r}; \mathbf{r}_0) = \mathbf{f}_o(\mathbf{r}_0) + \mathbf{J}_{\mathbf{q}}(\mathbf{r}_0)^T \, \Delta\mathbf{r} = \mathbf{0}, \tag{4.9}$$

   where $\mathbf{J}_{\mathbf{q}} = [\mathbf{J}_{\mathbf{qq}}, \mathbf{J}_{\mathbf{qp}}] = \nabla_{\mathbf{r}} \, \mathbf{f}_{\mathbf{q}}$.

In each SQP iteration, the solver finds a guess step $\mathbf{\Delta r}$ such that the linear constraint is approximately satisfied and the quadratic objective function is minimized. Depending on how the formulation handles the interaction between objective function and the constraints, different submethods can be considered. Next, we will briefly review the three most widely used approaches: *Lagrange multipliers*, *penalty methods* and *augmented Lagrangian*. For a more thorough dissertation on the topic, we refer the reader to the work by Nocedal and Wright [183].

Lagrange multipliers

The method of Lagrange multipliers is a standard strategy in mathematical optimization for finding the local minima of a function subject to constraints. This method is based on considering a set of auxiliary variables $\lambda \in \mathcal{R}^m$ called *Lagrange multipliers* and finding the critical points of the *Lagrangian function*:

$$\mathcal{L}(\Delta\mathbf{r}, \lambda; \mathbf{r}_0) = \tilde{h}(\Delta\mathbf{r}; \mathbf{r}_0) + \lambda^T \, \tilde{\mathbf{f}}_{\mathbf{q}}(\Delta\mathbf{r}; \mathbf{r}_0). \tag{4.10}$$

The critical points of the Lagrangian occur at saddle points. In the case of an equality-constrained QP problem, finding the solution is equivalent to solving a linear system whose matrix is generally indefinite of the kind

$$\begin{bmatrix} \mathbf{H} & \mathbf{J}_{\mathbf{q}}^T \\ \mathbf{J}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{r} \\ \lambda \end{bmatrix} = \begin{bmatrix} -\mathbf{g} \\ -\mathbf{f}_{\mathbf{q}} \end{bmatrix} \tag{4.11}$$

where the matrix and the right-hand side are evaluated at $\mathbf{r}_0$.

Penalty method

The penalty method replaces the constrained QP iteration with an unconstrained QP, adding an additional term to the objective function called penalty function. This function consists of a penalty parameter μ multiplied by a measure of the violation of constraints, commonly the squared norm. Consequently, each iteration minimizes the following quadratic function:

$$\tilde{g}(\Delta\mathbf{r}; \mathbf{r}_0) = \tilde{h}(\Delta\mathbf{r}; \mathbf{r}_0) + \frac{\mu}{2} \|\tilde{\mathbf{f}}_{\mathbf{q}}(\Delta\mathbf{r}; \mathbf{r}_0)\|^2, \tag{4.12}$$

which simply reduces to solving a linear system of the form

$$(\mathbf{H} + \mu \, \mathbf{J}_{\mathbf{q}}^T \mathbf{J}_{\mathbf{q}}) \, \Delta\mathbf{r} = -\mathbf{g} - \mu \, \mathbf{J}_{\mathbf{q}}^T \, \mathbf{f}_{\mathbf{q}}, \tag{4.13}$$

with all magnitudes evaluated at $\mathbf{r}_0$. In each iteration of the SQP, the penalty coefficient is successively incremented to guarantee the enforcement of the constraints. Ideally, the solution of these series of unconstrained problems would converge to the solution of the original constrained problem. However, finding the appropriate update scheme for the penalty coefficient is not trivial, and it highly determines the success of the algorithm. Additionally, taking $\mu \to \infty$ in late iterations may yield ill-conditioned system matrices.

Augmented Lagrangian

Similar to the penalty method, the augmented Lagrangian method also replaces the constrained SQP by a series of unconstrained QPs, by adding a penalty term to the objective function. However, to overcome the potential difficulties of the penalty method, the augmented Lagrangian adds an extra penalty term designed to mimic Lagrangian multipliers:

$$\tilde{g}(\Delta\mathbf{r};\mathbf{r}_0) = \tilde{h}(\Delta\mathbf{r};\mathbf{r}_0) + \frac{\mu}{2}\|\tilde{\mathbf{f}}_{\mathbf{q}}(\Delta\mathbf{r};\mathbf{r}_0)\|^2 + \lambda^T\,\tilde{\mathbf{f}}_{\mathbf{q}}(\Delta\mathbf{r};\mathbf{r}_0), \tag{4.14}$$

which, once again, reduces to solving a linear system of the form

$$(\mathbf{H} + \mu\,\mathbf{J}_{\mathbf{q}}^T\mathbf{J}_{\mathbf{q}})\,\Delta\mathbf{r} = -\mathbf{g} - \mu\,\mathbf{J}_{\mathbf{q}}^T\,\mathbf{f}_{\mathbf{q}} + \lambda^T\,\mathbf{J}_{\mathbf{q}}. \tag{4.15}$$

Here λ is an estimation of the Lagrangian multipliers, which is updated after each iteration as $\lambda \leftarrow \lambda - \mu\mathbf{f}_{\mathbf{q}}(\mathbf{r}^*)$, for a given candidate QP solution $\mathbf{r}^* = \mathbf{r}_0 + \Delta\mathbf{r}^*$. The resulting estimate improves at every step, progressively reducing the need for the penalty term. Consequently, it is not necessary to take $\mu \to \infty$, and thus ill-conditioned problems are avoided.

Practical considerations

There are some important aspects for the practical implementation of these methods:

1. **Box constraints**. Given the formulations presented above, it is easy to add box constraints on the design parameters $\mathbf{p}_m < \mathbf{p} < \mathbf{p}_M$ to each quadratic subproblem. One might simply project the resulting step within the bounds, but that does not necessarily guarantee that the solution is optimal and hence compromises the quadratic convergence of the SQP method. The literature on box-constrained QP problems is extensive and lies out of the scope of this text, but we suggest using the *gradient projection* method, see e.g., [183].

2. **Linear system solving**. Solving the linear systems associated to each of these formulations is not free of perils. A positive definite Hessian $\mathbf{H}$ is preferred to guarantee the convergence of the SQP algorithm. The regularization methods suggested in section 3.4 are also applicable in this context. However, it is also possible to use approximations to the full Hessian such that positive definiteness is satisfied. We will later describe these approaches in section 4.3.3 in the context of constraint exploration solutions.

3. **Step length selection**. Independently of the particular method used, each iteration results in a new guess step $\Delta\mathbf{r}$ and hence a candidate solution $\mathbf{r}_k = \mathbf{r}_{k-1} + \alpha\,\Delta\mathbf{r}$, which should be closer to a local minimum of the objective. Here, α represents an adapted step length. This is necessary because the QP is just an approximation of the more complex nonlinear problem, and its resulting step does not always lead to the improvement of the nonlinear objective

function h. Independently of the particular step-length selection method, e.g., line-search, trust-region, etc., it is necessary to define a merit function $\Psi(h, \mathbf{f})$ that measures the quality of a given solution. For an unconstrained problem, there is a straightforward solution, i.e., $\Psi = h$. However, in the constrained case there is no unique way of formulating this metric. Recall that most design problems have no perfect solution; therefore, objective minimization h and constraint fulfillment $\mathbf{f}_q = \mathbf{0}$ are opposing goals. A common choice is to use the penalty method function, $\Psi = h + \gamma \, ||\mathbf{f}||^2$, where γ is a weighting factor that adaptively changes depending on the convergence properties of the problem. γ should be small at the beginning of the optimization, and is progressively increased to ensure that the final solution is in static equilibrium.

Box constraints and the regularization of the linear system are standard issues in the resolution of QP subproblems. Therefore, from this point onward, we will ignore these issues and refer the reader to this section whenever a box-constrained QP problem is formulated.

Discussion

A few practical observation about the constrained SQP method:

1. At each iteration, the static equilibrium constraint may not hold for the candidate solution. This allows a broad exploration of the total space, which might facilitate avoiding local minima.

2. Appropriately selecting a valid merit function, as well as adaptively finding a compromise between the objective function h and constraint fulfillment $\mathbf{f_q} = \mathbf{0}$ is highly non-trivial.

3. When using Lagrange multipliers, the linear system (4.11) solved at each iteration might be relatively costly to solve. The system size is $t \times t$, with $t = 2n + m$. Additionally, the matrix $\mathbf{A}$ is usually indefinite; therefore, fast linear solvers for symmetric positive semi-definite matrices cannot be used.

4. When using penalty and augmented Lagrangian methods, tuning the penalty coefficient appropriately can be difficult and, as we have seen, leads to ill-conditioned system matrices.

5. When considering at the same time several k_p deformed poses with their own boundary conditions, these methods do not scale well. All the poses share the same parameter set and the formulation cannot be easily decoupled. For k_p poses, the size of the problems results $2k_p n + m$.

Except for the first observation, the rest constitute serious disadvantages on the applicability of the constrained SQP method in our particular scenario. In the following section, we introduce how we alternatively approach the problem (4.6), using sensitivity analysis to iteratively explore the equilibrium constraint looking for better solutions.

4.2.4 Constraint exploration solution

Some works in the literature follow alternative approaches for the solution of the constrained problem (4.6), based on the exploration of the constraint manifold, e.g., [14, 22, 23, 148, 68, 165]. Given some initial guess $\mathbf{r}_0$, such that $\mathbf{f}_\mathbf{q}(\mathbf{r}_0) = \mathbf{0}$, these methods iteratively compute a tentative step $\Delta\mathbf{r}$, such that the candidate solution $\mathbf{r}_1 = \mathbf{r}_0 + \Delta\mathbf{r}$ reduces the objective function h and approximately satisfies the constraint $\mathbf{f}_\mathbf{q} \simeq \mathbf{0}$. This same idea can be implemented using several numerical methods. Yang et al. [26], for instance, generate steps on the manifold considering both first and second order approximants –i.e., tangent space and quadratically parameterized osculant surface– that can be directly computed in the parameter space. Alternatively, we approximate the constraint manifold using a sensitivity analysis matrix computed by virtue of the implicit function theorem. In the next section, we will explain in detail how to solve the optimization problem (4.6) using sensitivity-based constraint exploration.

4.3 Sensitivity-based optimization

In this section, we explain in detail an alternative solution to the nonlinear constrained optimization, based on the exploration of the constraint manifold. Roughly, this method finds an explicit expression of the equilibrium manifold through local approximation, by virtue of sensitivity analysis. In the next section 4.3.1, we explain how this is done using the implicit function theorem, and we study the resulting implicit equilibrium function. Then, in section 4.3.2, we reformulate the problem (4.6) using this function, and we outline a SQP method for solving it. Finally, we will focus on two specific aspects of the algorithm that strongly determine its performance: the quadratic subproblem 4.3.3 and the projection to the equilibrium manifold 4.3.4.

4.3.1 Implicit equilibrium

We seek a mathematical expression of the the static equilibrium constraint manifold. The implicit function theorem states that if an equation $\mathbf{f}(q_1, \ldots, q_n, p_1, \ldots, p_m) = \mathbf{f}(\mathbf{q}, \mathbf{p}) = 0$ satisfies continuity and differentiability conditions, then there exists an implicit function $\hat{\mathbf{q}} : \mathcal{R}^m \to \mathcal{R}^n$ such that $\hat{\mathbf{q}}(\mathbf{p})$ locally approximates the locus defined by $\mathbf{f}(\mathbf{q}, \mathbf{p}) = 0$. That is, $\hat{\mathbf{q}}$ is an implicit function parameterized by $\mathbf{p}$ of the static equilibrium configuration as defined in (4.4). Although there might not exist a closed analytic expression of $\hat{\mathbf{q}}$, we can easily compute a linear approximation as follows. Assuming that the constraint $\mathbf{f_q}$ is a continuously differentiable function and there exists a point $(\mathbf{q}_0, \mathbf{p}_0)$ for which its value satisfies $\mathbf{f}(\mathbf{q}_0, \mathbf{p}_0) = \mathbf{0}$, we can apply the Taylor expansion series on the function $\mathbf{f_q}$, assuming that the implicit function $\hat{\mathbf{q}}(\mathbf{p})$ does exist:

$$\mathbf{f_q}(\hat{\mathbf{q}}(\mathbf{p}), \mathbf{p}) \Rightarrow \mathbf{f_q} \simeq \mathbf{f_q}(\mathbf{p}_0) + \frac{\mathbf{df_q}}{\mathbf{dp}}\Delta\mathbf{p} = \mathbf{0}. \tag{4.16}$$

We know $\mathbf{f_q}(\mathbf{p}_0) = \mathbf{0}$; therefore, applying the chain rule we have:

$$\frac{\mathbf{df_q}}{\mathbf{dp}}\Delta\mathbf{p} = \mathbf{0} \Rightarrow \mathbf{J_{qq}}\,\frac{\partial\hat{\mathbf{q}}}{\partial\mathbf{p}}\,\Delta\mathbf{p} + \mathbf{J_{qp}}\,\Delta\mathbf{p} = \mathbf{0}. \tag{4.17}$$

An finally, factoring $\Delta\mathbf{p}$ out and reordering terms:

$$\mathbf{J_{qq}}\,\frac{\partial\hat{\mathbf{q}}}{\partial\mathbf{p}} + \mathbf{J_{qp}} = \mathbf{0} \Rightarrow \mathbf{J_{qq}}\,\frac{\partial\hat{\mathbf{q}}}{\partial\mathbf{p}} = -\mathbf{J_{qp}}. \tag{4.18}$$

The matrix resulting from solving the linear system,

$$\mathbf{S} = \frac{\partial\hat{\mathbf{q}}}{\partial\mathbf{p}}, \tag{4.19}$$

is called *sensitivity matrix*, and the corresponding linear approximation of the implicit function, $\hat{\mathbf{q}} \simeq \mathbf{q}_0 + \mathbf{S}\Delta\mathbf{p}$, defines a linear manifold that is locally tangent to the equilibrium manifold. This can be easily proved by substituting $\Delta\mathbf{q} = \mathbf{S}\Delta\mathbf{p}$ in the equation that defines the null-space of the equilibrium constrain:

$$\begin{bmatrix}\mathbf{J_{qq}} & \mathbf{J_{qp}}\end{bmatrix}\begin{bmatrix}\Delta\mathbf{q}\\ \Delta\mathbf{p}\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix}, \tag{4.20}$$

which holds true independently of the value of $\Delta\mathbf{p}$.

Sensitivity matrix

The sensitivity matrix spans the plane tangent to the equilibrium constraint at the evaluation point $\mathbf{r}_0$. We will refer to the resulting linear manifold $\tilde{\mathcal{E}}$ as the

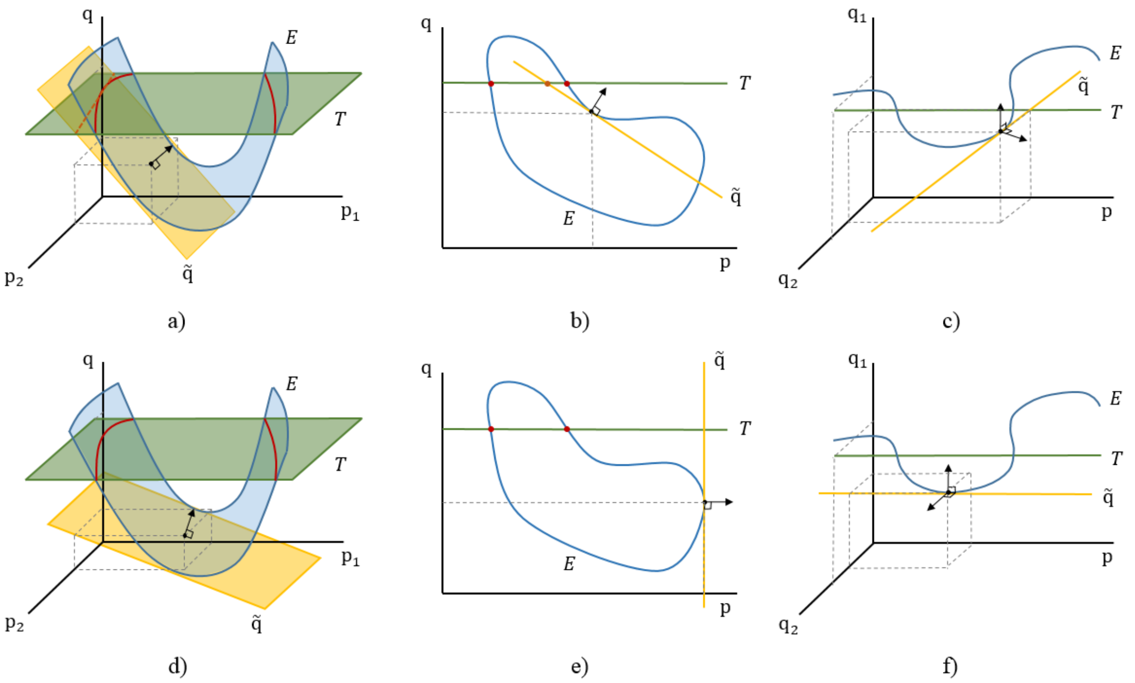


Figure 4.2: This picture shows linear approximations to the equilibrium manifold, for the different situations that were described in section 4.1.2. The top row depicts some good approximation examples. However, depending on the nonlinearities of the equilibrium manifold, the linear manifold might produce poor estimates (bottom left) or degenerate cases with infinite (bottom center) or singular (bottom right) functions.

linear equilibrium manifold. Fig. 4.2 shows the linear equilibrium approximations corresponding to the three design situations introduced in section 4.1.2. In practical terms, the sensitivity matrix allows us to compute, for a given parameter change $\Delta\mathbf{p}$, the displacement in the deformed configuration $\Delta\mathbf{q}$ such that the linearized static equilibrium constraint $\tilde{\mathbf{f}}_{\mathbf{q}} = \mathbf{0}$ holds. Some other useful conclusions can be drawn from the numerical properties of the matrix:

- **Linear equilibrium**: In general, the sensitivity matrix $\mathbf{S}$ has dimension $n \times m$ and $[\mathbf{S}^T, \mathbf{I}]$, for $\mathbf{I}_{m\times m}$, is a basis whose rows span the linear equilibrium m-manifold. The intersection of this linear manifold and the target manifold, if it exists, approximates the set of perfect solutions to the problem introduced in section 4.1.2 (Fig. 4.2, top row).

- **Infinite sensitivity**: If $\mathbf{J}_{\mathbf{qq}}$ is rank-deficient, the linear system in (4.18) cannot be solved. This corresponds to situations for which the linear equilibrium approximation runs parallel to the shape space Q –see Fig. 4.2 picture *e*. In practical terms, this means that the deformed configuration is infinitely sensitive w.r.t. some of the design parameters, and the matrix is not valid.

- **Singular sensitivity**: If $\mathbf{J_{qp}}$ is rank-deficient, the linear system in (4.18) can be solved, but the resulting sensitivity matrix is also rank-deficient. This corresponds to points where there exists at least a direction in the design space P for which the deformed configuration does not change. Here, the sensitivity matrix is still valid but it may produce difficult situations when solving the optimization. In the worst case scenario, the linear equilibrium approximation runs parallel to the design space P –see Fig. 4.2, picture f. In such case, traversing the linear equilibrium does not change the objective function.
- **Ill-conditioning**: Sometimes, $\mathbf{J_{qq}}$ and $\mathbf{J_{qp}}$ matrices are ill-conditioned or numerically close to being rank-deficient. This might cause that $\mathbf{S}$ is not well-behaved in two ways. On one hand, the condition number –i.e., the difference between minimum and maximum singular values– may be high, what means that the deformed configuration is much more sensitive to some design parameters than others. On the other hand, the linear equilibrium manifold may be almost parallel to the target manifold, what means its navigation have very little effect on reducing the objective function.

In general, the sensitivity matrix provides us with a formal framework to rewrite the optimization problem (4.6) in terms of the implicit function $\hat{\mathbf{q}}$, as we will see in the next section.

4.3.2 Sensitivity-based SQP

The implicit equilibrium $\hat{\mathbf{q}}(\mathbf{p})$ defines the equilibrium configuration corresponding to a given parameter vector $\mathbf{p}$. Using this function, we can remove the static equilibrium constraint in (4.6), and reformulate the objective function h to evaluate shape descriptors only on equilibrium configurations:

$$\hat{h} = \frac{1}{2} \parallel \hat{\mathbf{q}}(\mathbf{p}) - \mathbf{q}^t \parallel^2. \tag{4.21}$$

This way, the final equilibrium shape optimization problem is:

$$\begin{aligned} &\min_{\mathbf{p}} \hat{h} \\ &\text{s.t. } \mathbf{p}_m \leq \mathbf{p} \leq \mathbf{p}_M. \end{aligned} \tag{4.22}$$

Note that the force equilibrium constraint is no longer needed as we implicitly consider only deformed configurations in static equilibrium $\hat{\mathbf{q}}$. Generally, there is no explicit analytic expression for this magnitude, which can be highly nonlinear, and one can only rely on locally tangent approximations $\tilde{\mathbf{q}}$ –e.g., the one defined in the previous section based on the sensitivity matrix. This problem finds minima at:

$$\hat{\mathbf{g}} = \frac{\partial \hat{h}}{\partial \mathbf{p}} = \mathbf{0} \Rightarrow (\mathbf{q} - \mathbf{q}^t)^T \frac{\partial \hat{\mathbf{q}}}{\partial \mathbf{p}} = \mathbf{0}. \tag{4.23}$$

Algorithm 1: Sensitivity-based SQP

```
Data: Target shape value: $\mathbf{q}^t$
Data: Initial guess: $\{(\mathbf{q}_0, \mathbf{p}_0) \mid \mathbf{f}_\mathbf{q}(\mathbf{q}_0, \mathbf{p}_0) = \mathbf{0}\}$
Data: Gradient and constraints tolerance: $t_g, t_f$
Result: Optimal solution:
        $\{(\mathbf{q}^*, \mathbf{p}^*) \mid \|\mathbf{f}_\mathbf{q}(\mathbf{q}^*, \mathbf{p}^*)\| \le t_f$ and $\|\hat{\mathbf{g}}(\mathbf{q}^*, \mathbf{p}^*, \mathbf{q}^t)\| \le t_g\}$
 1. Assign initial solution $(\mathbf{q}_k, \mathbf{p}_k) \leftarrow (\mathbf{q}_0, \mathbf{p}_0)$ ;
while $\| \hat{\mathbf{g}}(\mathbf{q}_k, \mathbf{p}_k, \mathbf{q}^t) \| > t_g$ do
    2. Quadratically approximate objective $\tilde{h}(\mathbf{p}; \mathbf{r}_k) \simeq \hat{h}(\hat{\mathbf{q}})$ ;
    3. Solve QP subproblem: $\tilde{\mathbf{p}}_{k+1} \leftarrow$ SOLVE$(\tilde{h}, \mathbf{p}_m, \mathbf{p}_M)$ ;
    4. Compute the implicit deformation $\tilde{\mathbf{q}}_{k+1} = \tilde{\mathbf{q}}(\tilde{\mathbf{p}}_{k+1})$ ;
    5. Compute candidate steps: $\Delta\tilde{\mathbf{q}} = \tilde{\mathbf{q}}_{k+1} - \mathbf{q}_k$ and $\Delta\tilde{\mathbf{p}} = \tilde{\mathbf{p}}_{k+1} - \mathbf{p}_k$ ;
    do
        6. Compute candidate solutions: $\tilde{\mathbf{q}}_{k+1} \leftarrow \mathbf{q}_k + \Delta\tilde{\mathbf{q}}$, $\tilde{\mathbf{p}}_{k+1} \leftarrow \mathbf{p}_k + \Delta\tilde{\mathbf{p}}$ ;
        7. Recompute equilibrium $(\mathbf{q}_{k+1}, \mathbf{p}_{k+1}) \leftarrow$ PROJECT$(\tilde{\mathbf{q}}_{k+1}, \tilde{\mathbf{p}}_{k+1}, t_f)$ ;
        8. Reduce candidate steps length: $\Delta\tilde{\mathbf{q}} \leftarrow \alpha\,\Delta\tilde{\mathbf{q}}$ and $\Delta\tilde{\mathbf{p}} \leftarrow \alpha\,\Delta\tilde{\mathbf{p}}$ ;
    while $(h(\mathbf{q}_{k+1}, \mathbf{p}_{k+1})) < h(\mathbf{q}_k, \mathbf{p}_k))$;
end
```

The resulting optimization is a nonlinear least squares problem. The algorithm 1 outlines the basic steps for the resolution of the optimization problem (4.22) using a standard unconstrained SQP method with a simple line-search procedure for step length selection. There are three key parts of the algorithm that will define its performance:

1. **Quadratic subproblem solution (Steps 2–3)**. In each iteration of the algorithm, the nonlinear objective function (4.21) is approximated with a quadratic function, which turns the minimization into a standard box-constrained QP problem. The subroutine SOLVE encapsulates the computation of a solution for such problem. Depending on the type of approximation to the objective function that is used, the overall SQP convergence properties vary. As we will see, this is also related with how the implicit equilibrium function is estimated $\tilde{\mathbf{q}}(\mathbf{p}; \mathbf{r}_k) \simeq \hat{\mathbf{q}}$. In section 4.3.3, we go deeper into the relation between QP methods and equilibrium manifold estimation.

2. **Static equilibrium computation (Step 7)**. The implicit equilibrium function estimation $\tilde{\mathbf{q}}(\mathbf{p}; \mathbf{r}_k)$ is only locally tangent to the static equilibrium constraint. Solving the quadratic subproblem will produce a candidate step

$(\Delta\tilde{\mathbf{q}}, \Delta\tilde{\mathbf{p}})$ that lies in the manifold defined by the implicit equilibrium approximation but for which the static equilibrium constraint does not necessarily hold. The longer the step, the larger the deviation from the equilibrium manifold $\| \, \mathbf{f}_q(\mathbf{q}_k + \Delta\tilde{\mathbf{q}}, \mathbf{p}_k + \Delta\tilde{\mathbf{p}}) \, \| > 0$. The objective function is defined only in terms of a deformed configuration in static equilibrium. For that reason, after each quadratic step, the solution must be projected back to the equilibrium manifold using the subroutine PROJECT. We have already explained how to compute the static equilibrium deformation for a given constant parameter vector in section 3.4. In section 4.3.4, we introduce an alternative solution scheme that is advantageous in some situations.

3. **Step length selection (Steps 6–8)**. Minimizing the quadratic approximation of the objective function $\tilde{h}(\mathbf{p}; \mathbf{r}_k)$ reduces the nonlinear objective h for a sufficiently small step. In this algorithm outline, we consider the most basic line-search procedure: the length of the step is iteratively bisected until the objective function h is reduced. Adequately selecting the step length is critical for our problem, as each time a new candidate solution is evaluated, it is necessary to compute the corresponding static equilibrium, what might be costly. The study of alternative step selection schemes, e.g., Armijo line-search or trust-region methods, goes beyond the scope of this thesis. For more information, we refer the reader to [183].

4.3.3 Quadratic subproblem

The standard quadratic approximation of the objective function (4.21) is:

$$\hat{h} \simeq \tilde{h} = \frac{1}{2} \Delta\mathbf{p}^T \, \mathbf{H} \, \Delta\mathbf{p} + \Delta\mathbf{t}^T \, \mathbf{g} + c, \tag{4.24}$$

where the vector $\mathbf{g} = \nabla_{\mathbf{p}} \hat{h}$ is the gradient of the objective function, the matrix $\mathbf{H} = \nabla^2_{\mathbf{pp}} \hat{h}$ is the Hessian, $\Delta\mathbf{p} = \mathbf{p}_{k+1} - \mathbf{p}_k$, $\Delta\mathbf{t} = \mathbf{q}^t - \mathbf{q}_k$, and c is a scalar constant that does not affect the result of the minimization. Computing the critical point w.r.t. $\Delta\mathbf{p}$ of this quadratic function reduces to solving the linear system $\mathbf{H} \, \Delta\mathbf{p} = -\mathbf{g}$. As long as the system matrix is positive definite, the resulting step lies on a descent direction of the objective function $\hat{h}$, and the step $\alpha\Delta\mathbf{p}$ improves the nonlinear solution for a sufficiently small value of α.

Quadratic subproblems are completely characterized by the approximation to the Hessian and gradient. Depending on the particular approximation of the Hessian, the overall SQP convergence properties vary. As we will see, this is also related with how the implicit equilibrium function $\tilde{\mathbf{q}}(\mathbf{p}; \mathbf{r}_k) \simeq \hat{\mathbf{q}}$ is estimated. In the following section, we review the most common approximations that have been used in the literature, and we illustrate their geometrical interpretation.

Linear approximation (Gauss-Newton step)

The simplest estimation of the implicit equilibrium function $\hat{\mathbf{q}}$ is the linear approximation, which uses the sensitivity matrix previously introduced in section 4.3.1:

$$\hat{\mathbf{q}} \simeq \tilde{\mathbf{q}}(\mathbf{p}; \mathbf{r}_k) = \mathbf{q}_k + \mathbf{S}\,\Delta\mathbf{p}. \tag{4.25}$$

Using this estimation, the quadratic objective function becomes:

$$\hat{h} \simeq \tilde{h} = \frac{1}{2}\,||\mathbf{q}_0 + \mathbf{S}\,\Delta\mathbf{p} - \mathbf{q}^t||^2, \tag{4.26}$$

which in the standard form defined in equation (4.24) is

$$\hat{h} \simeq \tilde{h} = \frac{1}{2}\,\Delta\mathbf{p}^T\,\mathbf{S}^T\mathbf{S}\,\Delta\mathbf{p} + \Delta\mathbf{t}^T\,\mathbf{S}\,\Delta\mathbf{p} + \Delta\mathbf{t}^T\Delta\mathbf{t}, \tag{4.27}$$

and the corresponding gradient and Hessian are

$$\mathbf{g} = \mathbf{S}^T\,\Delta\mathbf{t}, \qquad \mathbf{H} = \mathbf{S}^T\mathbf{S}. \tag{4.28}$$

This formulation is equivalent to the so-called Gauss-Newton step and the resulting Hessian $\mathbf{H}$ is symmetric and semi-positive definite by definition. The geometric interpretation of this solution follows straightforwardly from figure Fig. 4.2, where linear approximations to the equilibrium constraint manifold are depicted. The target $\mathcal{T}$ and equilibrium $\mathcal{E}$ manifolds become linear functions. The problem reduces to the computation of the shortest distance between two linear manifolds, and it can be analyzed in terms of the dimensionality of these manifolds:

- $\mathbf{Dim}(\mathcal{P}) > \mathbf{Dim}(\mathcal{S})$. This is analogous to the intersection of two planes in affine 3D space, which has infinite solutions as shown in Fig. 4.2, picture *a*. In this situation, $\mathbf{S}$ has dimension $n \times m$ with $m >> n$; therefore, $\mathbf{H} = \mathbf{S}^T\mathbf{S}$ is generally singular. The step can be computed by solving

  $$\min_{\Delta\mathbf{p}} \frac{1}{2}\Delta\mathbf{p}^T\Delta\mathbf{p} \quad \text{s.t} \quad \mathbf{H}\Delta\mathbf{p} = -\mathbf{g},$$

  what selects the solution with the smallest norm.

- $\mathbf{Dim}(\mathcal{P}) = \mathbf{Dim}(\mathcal{S})$. This is analogous to the intersection of two lines in affine 2D space, which has a single solution as shown in Fig. 4.2, picture *b*. In this situation, $\mathbf{S}$ has dimension $n \times n$; therefore, $\mathbf{H} = \mathbf{S}^T\mathbf{S}$ is generally invertible.

- **Dim**$(\mathcal{P})$ < **Dim**$(\mathcal{S})$. This is analogous to the intersection of two lines in affine 3D space, which has generally no solution as shown in Fig. 4.2, picture *c*. In this situation, $\mathbf{S}$ has dimension $n \times m$ with $m < n$; therefore $\mathbf{H} = \mathbf{S}^T\mathbf{S}$ is generally invertible. The problem is equivalent to the one of finding the shortest distance between two lines.

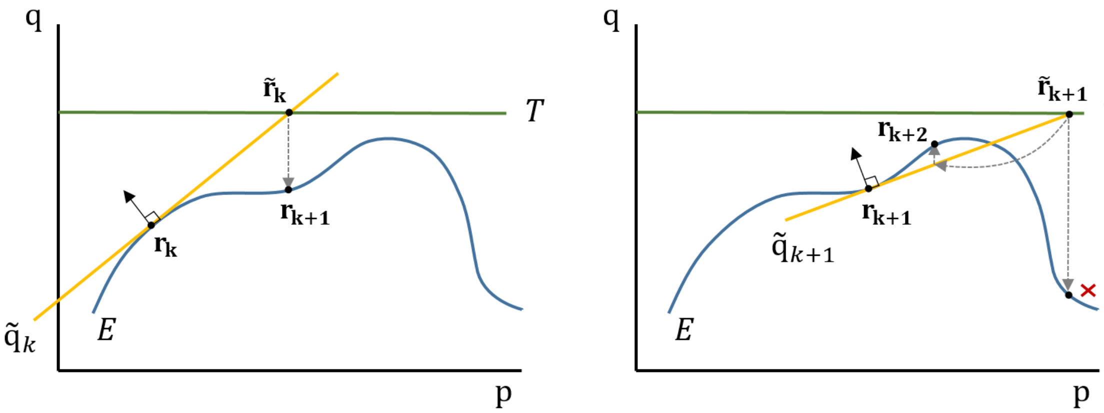


Figure 4.3: This picture shows two iterations of Gauss-Newton steps for a 2D example problem. Left) in the first iteration, the method provides a good approximation. Right) in the second iteration, the method clearly overshoots the solution, and the candidate step requires a bisection to guarantee convergence.

In section 4.3.1, we introduced potential problems connected with the properties of the sensitivity matrix $\mathbf{S}$. An interesting geometric observation is that if $\mathbf{S}$ is almost singular, ill-conditioned, or has relatively small norm, the linear equilibrium and target manifolds might be close to parallel in some direction of the design space P. If the estimation point $\mathbf{r}_k$ is still far from the optimum, then the calculation of the closest point between the linear equilibrium and target manifolds would result in large steps, as depicted in Fig. 4.3 (right). These steps often overshoot the disc for which the quadratic function $\tilde{h}$ is a good estimator of the nonlinear function $\hat{h}$ and, consequently, they yield poor performance due to the costly line-search. In such cases, it might be advisable to use SQP schemes with a more conservative step length selection procedure, e.g., the Levenberb-Marquardt trust-region algorithm.

Quadratic approximation (Newton-Raphson step)

To reduce the problem of overshooting the step, a higher-order expression could be used to estimate the implicit equilibrium $\hat{\mathbf{q}}$. Its quadratic approximation yields:

$$\hat{\mathbf{q}} \simeq \tilde{\mathbf{q}}(\mathbf{p};\mathbf{r}_k) = \mathbf{q}_k + \mathbf{S}\,\Delta\mathbf{p} + \frac{1}{2}\,\Delta\mathbf{p}^T\,\mathbf{C}\,\Delta\mathbf{p}, \tag{4.29}$$

where $\mathbf{C}$ is the third-order tensor of second derivatives $\mathbf{C} = \nabla^2_{\mathbf{p}}\mathbf{q}$, which indicates the curvature of the implicit function $\hat{\mathbf{q}}$. Using this estimation, the corresponding objective function is:

$$\hat{h} \simeq \tilde{h} = \frac{1}{2}\,\|\mathbf{q}_k + \mathbf{S}\,\Delta\mathbf{p} + \frac{1}{2}\,\Delta\mathbf{p}^T\,\mathbf{C}\Delta\mathbf{p} - \mathbf{q}^t\|^2, \tag{4.30}$$

which in the standard form defined in equation (4.24) is

$$\hat{h} \simeq \tilde{h} = \frac{1}{2}\,\Delta\mathbf{p}^T\,(\mathbf{S}^T\mathbf{S} + \Delta\mathbf{t}^T\mathbf{C})\,\Delta\mathbf{p} + \Delta\mathbf{t}^T\,\mathbf{S}\,\Delta\mathbf{p} + \Delta\mathbf{t}^T\Delta\mathbf{t}. \tag{4.31}$$

Here, third and fourth order terms have been removed to consider only a quadratic approximation. The corresponding gradient and Hessian of the QP problem are:

$$\mathbf{g} = \mathbf{S}^T\,\Delta\mathbf{t}, \qquad \mathbf{H} = \mathbf{S}^T\mathbf{S} + \Delta\mathbf{t}^T\mathbf{C}. \tag{4.32}$$

This formulation is equivalent to the so-called Newton-Raphson step, resulting from the direct application of the quadratic Taylor expansion to the nonlinear objective function $\hat{h}$. Fig. 4.4 shows a depiction of Newton step in the same scenario presented above. Note that although the expression (4.30) uses a quadratic approximation $\tilde{\mathbf{q}}$ of the implicit equilibrium function $\hat{\mathbf{q}}$, the QP step is computed minimizing a distance where the third and fourth order terms have been removed. Consequently, the QP problem does not exactly minimize the distance from the target manifold to a quadratic expression of the implicit equilibrium, but an approximation.

Also note that the Gauss-Newton problem formulated in (4.27) differs from the Newton-Raphson problem formulated in (4.31) only in the expression for the Hessian. Gauss-Newton misses the term with second derivatives of the implicit equilibrium manifold $\mathbf{C}$. However, as with Gauss-Newton, there are also reasons for not using the standard Hessian in Newton's method:

- Analytically computing the second derivatives of the implicit equilibrium manifold $\mathbf{C}$ is generally very costly. It is possible to compute a numerical estimation of this term using finite differences, but that requires solving m static equilibria plus m sensitivity computations, one per differential change in $\mathbf{p}$.
- The term $\mathbf{C}$ in (4.31) is scaled by $\Delta\mathbf{t}$, which is the current difference vector to the solution $\mathbf{q}^t$. Consequently, the influence of this term on the Hessian is progressively smaller as the solution converges to the optimum. In practical

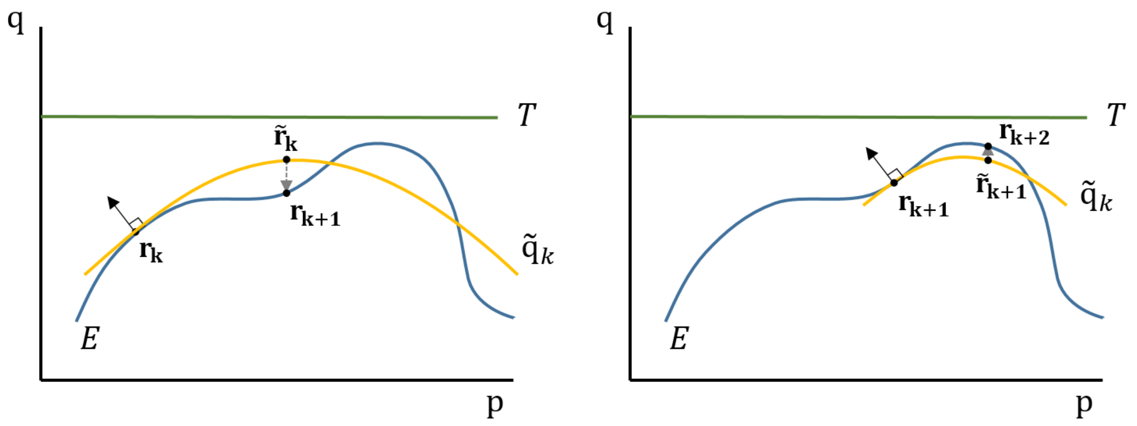


Figure 4.4: This picture shows the example depicted in Fig. 4.3 using Newton-Raphson steps. In both cases, the higher-order estimation of the equilibrium manifold leads to better candidate solutions and improved convergence.

terms, for a problem where the equilibrium and target manifolds are not far apart, the term depending on the first-order approximation $\mathbf{S}$ will suffice for a good convergence.

- There is no guarantee that the complete Hessian $\mathbf{H} = \mathbf{S}^T\mathbf{S} + \Delta\mathbf{t}^T\mathbf{C}$ is definite positive. Indefinite matrices can produce non-descendent steps when solving the linear system associated to the QP problem. In general, indefinite Hessians arise when there are saddle points in the objective function $\hat{h}$, which is relatively common considering that the equilibrium manifold is often highly nonlinear.

To avoid some of the limitations mentioned above, many classic works on optimization have proposed alternative approximations to the Hessian, mainly based on rank-one iterative updates. Next, we will discuss two of them.

Quasi-Newton steps

Quasi-Newton methods are a class of optimization algorithms characterized by approximating the Hessian with a matrix that fulfills certain properties. Quasi-Newton methods are often used when the full Hessian is unavailable, too expensive, or ill-behaved. In general, the approximation is iteratively computed as $\mathbf{B}_k = \mathbf{B}_{k-1} + \mathbf{B}$ such that the *secant equation* is satisfied. Considering the gradient $\mathbf{g}$ and the parameters $\mathbf{p}$ in the current and previous steps, the secant equation is stated as $\mathbf{y}_k = \mathbf{B}\,\mathbf{s}_k$, for $\mathbf{y}_k = \mathbf{g}(\mathbf{p}_k) - \mathbf{g}(\mathbf{p}_{k-1})$, and $\mathbf{s}_k = \mathbf{p}_k - \mathbf{p}_{k-1}$. If the dimensionality of the problem $m > 1$, the secant equation is underdetermined. The various quasi-Newton methods are formulated on the basis of the choice of constraints imposed to the resolution of the secant equation, which in turn lead to different one-rank updates. We follow the well-known subclass called BFGS (*Broyden-Fletcher-Goldfarb-Shanno*), which guarantees that the resulting matrix is symmetric and positive definite. Given that $\mathbf{y}_k^T \mathbf{s}_k > t_b$ holds, for some tolerance t_b, two alternative updates can be used:

1. **Basic BFGS version**. Starting from $\mathbf{B}_0 = \mathbf{I}$, the following formula computes an update $\mathbf{B}$ to the approximation of the full Hessian, $\mathbf{B}_k = \mathbf{B}_{k-1} + \mathbf{B} \simeq \mathbf{H}_k$, such that it remains symmetric and positive definite:

$$\mathbf{B} = \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k} - \frac{\mathbf{B}_k \mathbf{s}_k \mathbf{s}_k^T \mathbf{B}_k}{\mathbf{s}_k^T \mathbf{B}_k \mathbf{s}_k}. \tag{4.33}$$

2. **Gauss-Newton version**. Starting from $\mathbf{B}_0 = \mathbf{S}_0^T \mathbf{S}_0$ and $\mathbf{D}_0 = \mathbf{0}$, the following formula computes an update $\mathbf{D}$ only to the approximation of the third order term $\Delta \mathbf{t}^T \mathbf{C}$ introduced in (4.30), with

$$\mathbf{D} = \frac{\mathbf{z}_k \mathbf{y}_k^T + \mathbf{y}_k \mathbf{z}_k^T}{\mathbf{y}_k^T \mathbf{s}_k} - \frac{\mathbf{z}_k^T \mathbf{s}_k}{(\mathbf{y}_k^T \mathbf{s}_k)^2} \mathbf{y}_k \mathbf{y}_k^T, \tag{4.34}$$

where $\mathbf{z}_k = (\mathbf{y}^{\#} - \mathbf{D}_k \mathbf{s}_k)$ and $\mathbf{y}_k^{\#} = \mathbf{S}_k^T \Delta \mathbf{t}_k - \mathbf{S}_{k-1}^T \Delta \mathbf{t}_k$. Then, the approximation to the full Hessian is iteratively updated as $\mathbf{B}_k = \mathbf{S}_k^T \mathbf{S}_k + \mathbf{D}_k \simeq \mathbf{H}_k$, with $\mathbf{D}_k = \mathbf{D}_{k-1} + \mathbf{D}$.

Note that the latter option is specially attractive as it takes advantage of the already computed sensitivity matrix $\mathbf{S}$ and only refines the third order term iteratively, while keeping the Hessian well-behaved. In both cases, as an additional feature, BFGS offers the possibility of estimating either the Hessian $\mathbf{H}$, or directly its inverse $\mathbf{H}^{-1}$ using the *Sherman-Morrison* formula. For further details on the numerical properties of quasi-Newton methods, we refer the reader to Nocedal and Wright [183]. Note that, from a geometric perspective, the quasi-Newton step originates from the distance to a quadratic approximation of the implicit equilibrium in (4.30), in which the third and fourth order terms have been neglected. Consequently, the same limitation mentioned above for the Newton-Raphson step applies.

Additional note about performance

The computation of the sensitivity matrix $\mathbf{S}$ requires solving m linear systems, what might exceed the time-budget of an interactive design tool, like the one we present in chapter 6. For the purpose of analysis, throughout this chapter, we have formulated the sensitivity matrix explicitly. However, solving the QP subproblem requires only evaluating the gradient $\mathbf{g}$ and Hessian $\mathbf{H}$ of the objective function, and hence we could apply the *adjoint method* instead, and avoid computing $\mathbf{S}$ explicitly.

4.3.4 Equilibrium projection

Steps resulting from the solution of the QP subproblem satisfy the equilibrium constraint only approximately. That is, new candidate solutions satisfy $\tilde{\mathbf{f}}_{\mathbf{q}}(\tilde{\mathbf{q}}_{k+1}, \tilde{\mathbf{p}}_{k+1}) = \mathbf{0}$, only for some approximation $\tilde{\mathbf{f}}_{\mathbf{q}} \simeq \mathbf{f}_{\mathbf{q}}$, i.e., in the cases that we have considered, linear or quadratic. It is necessary to project each candidate solution back to the equilibrium manifold for two reasons:

1. The evaluation of the objective function is defined only for deformed configurations in static equilibrium. Therefore, to assess whether or not there has been an improvement in the nonlinear minimization for a given QP candidate solution, it is necessary to recompute the static equilibrium.

2. The implicit function theorem used for the estimation of $\hat{\mathbf{q}}$ in the next step assumes the hypothesis $\mathbf{f}_{\mathbf{q}} = \mathbf{0}$. Therefore, it is mandatory to perform manifold projection at the beginning of each step.

In this part of the thesis we analyze possible implementations of such projection subroutine. First, let us formalize some concepts that will be revisited in the following sections. Fig. 4.5 shows a simplified representation of the total space, where all variables corresponding to the deformed Q and design P subspaces are respectively represented in the $\mathbf{q}$ and $\mathbf{p}$ axes. We have thoroughly studied the constraint $\mathbf{f}_{\mathbf{q}} = \mathbf{0}$, which represents the m-manifold of static equilibrium configurations. However, as introduced at the beginning of this chapter, $\mathbf{f}_{\mathbf{p}} = \mathbf{0}$ defines another n-manifold at which the partial derivatives of the elastic potential w.r.t. design parameters vanish. Recall from section 3.4 that the minimization of potential energy leads to equilibrium constraint satisfaction. In this context, we might formulate the projection subroutine in three possible ways:

1. **Potential energy minimization in** R (Fig. 4.5, left): $\min_{\mathbf{r}} V(\mathbf{r})$.

2. **Potential energy minimization in** Q (Fig. 4.5, center): $\min_{\mathbf{q}} V(\mathbf{r})$.

3. **Root-finding in** R (Fig. 4.5, right): $\text{find}(\mathbf{r}) \quad \text{such that} \quad \mathbf{f}_{\mathbf{q}}(\mathbf{r}) = \mathbf{0}$

In the following section, we will consider each of this options and analyze their advantages in terms of flexibility, robustness and performance, for the particular case of inverse elastic design problems solved through constraint exploration.

Potential energy minimization in R

Even though this approach is not practical, as we will discuss next, it deserves some attention and analysis. Intersections between the two constraints $\mathbf{f}_q = 0$ and $\mathbf{f}_p = 0$

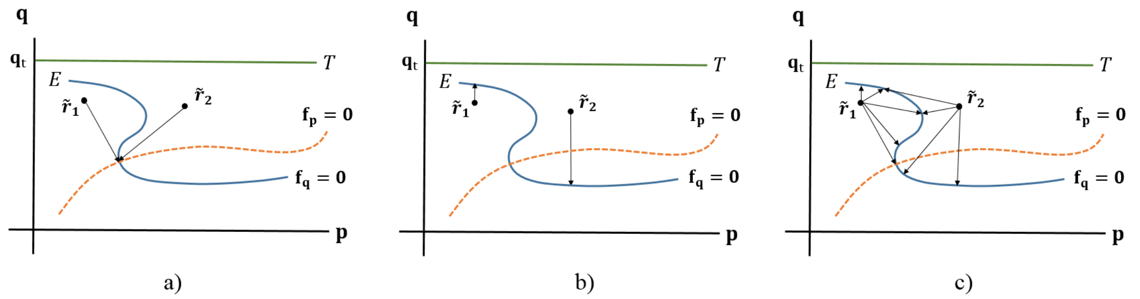


Figure 4.5: This picture shows a compact representation of the projection problem in the total space R, where all the DoFs corresponding to deformed Q and design P spaces have been condensed in the $\mathbf{q}$ and $\mathbf{p}$ axes. Three possible projection schemes are shown, for two different starting points $\tilde{\mathbf{r}}_1$ and $\tilde{\mathbf{r}}_2$: a) potential energy minimization in total space R finds the critical point of the potential energy function V w.r.t. $\mathbf{q}$ and $\mathbf{p}$; b) potential energy minimization in deformed space Q finds the critical point of V w.r.t. $\mathbf{q}$; and c) root-finding in total space R solves the nonlinear system of equations $\mathbf{f_q}(\mathbf{q}, \mathbf{p}) = \mathbf{0}$.

define critical points of the elastic potential function w.r.t. all the variables of the total space R. It would be possible to implement the projection by computing the minimization $\min_{\mathbf{p},\mathbf{q}} V(\mathbf{r})$ s.t. $\mathbf{p}_m \leq \mathbf{p} \leq \mathbf{p}_M$. However, this idea might not be advisable for two reasons:

- First, we are only interested on satisfying static equilibrium constraints on deformation variables, other considerations might deviate projection from the original goal and produce worse performance results.

- Second, the behavior of the potential energy function in the total space R is usually full of perils that lead to numerically difficult situations such as indefinite and stationary points. For instance, deformed and material configurations often produce opposite effects on elastic strains, what can easily lead to R having a non-empty null-space and hence an infinite number of solutions. Similarly, material parameters often lead to trivial solutions if they are not bounded –e.g., lower elastic stiffness produce always smaller potentials. Something similar happens with the gravitational potential energy, which vanishes if the undeformed configuration of the object collapses.

In practical terms, there is no straightforward application, in the inverse elastic design scenario, for finding the design space configuration with minimum internal energy. In addition, there might be too difficult or even impossible for the minimization to converge using standard procedures, making this option not worthy.

Potential energy minimization in Q

Intersections between the equilibrium manifold $\mathcal{E}$ and a fixed-parameter subspace $\mathbf{p} = \mathbf{p}_c$, define critical points of the elastic potential function w.r.t. the variables of the deformed space Q. The problem is reduced to solving the minimization problem $\min_{\mathbf{q}} V(\mathbf{r})$. Note that, even though the potential energy is affected by both deformed configuration $\mathbf{q}$ and design parameters $\mathbf{p}$, this optimization minimizes only the elastic potential w.r.t the deformed configuration. If we restrict to the deformed space Q, the potential energy is often relatively well-behaved. A robust SQP algorithm for the solution of this optimization problem has already been defined in section 3.4, together with various techniques to improve the convergence and robustness of the method against indefinite and stationary points.

Fig. 4.5 (center) shows the geometric interpretation of this particular projection subroutine. In general, the method produces good projected solutions (point $\mathbf{r}_1$). However, it is possible to find scenarios for which the equilibrium point is relatively far from the initial point in deformed space Q, compared to the corresponding distance in the total space R (point $\mathbf{r}_2$). This might have two main disadvantages:

1. Convergence to the equilibrium point may require more iterations of the optimization. As it has been mentioned before, a slow projection subroutine has a big impact on the performance of constraint exploration optimization methods, as it is sometimes necessary to evaluate the objective function –i.e., recompute projection– several times during step length selection.

2. In the specific case of sensitivity-based optimization methods, the candidate solution to project will be potentially close to a target configuration $\mathbf{q}^t$. Long projection distances would lead to deformed configurations far from the target, and hence slow down the overall convergence of the design optimization problem.

Motivated by these facts, in the next section we discuss an additional formulation, based on using a root-finding algorithm to solve a nonlinear system of equations, that might outperform the usual solution in some scenarios.

Root-finding in total space R

Fig. 4.5 (right) shows that, given a candidate solution in total space $\tilde{\mathbf{r}}_k$, there is actually an infinite number of possible projection points if we consider traversing the total space R instead of just the deformed space Q. Many of these options produce projected solutions that i) are closer to the initial candidate solution and/or ii) are closer to the target deformed configuration. We seek an algorithm to traverse the total space R and robustly converge to any of the solutions defined by the constraint $\mathbf{f_q} = \mathbf{0}$, subject to some performance criterion we might need depending on the particular application. In this section, we propose an approach based on the under-determined Newton algorithm. We can define the problem at hand analogously to the formulation of the static equilibrium introduced in section 3.4 based on Newtonian mechanics. We consider the following root finding problem:

$$\text{find}(\mathbf{r}) \quad \text{s.t.} \quad \mathbf{p}_m \leq \mathbf{p} \leq \mathbf{p}_M \quad \text{such that} \quad \mathbf{f_q}(\mathbf{r}) = \mathbf{0}, \tag{4.35}$$

where $\mathbf{f_q} : \mathcal{R}^o \to \mathcal{R}^n$ is a continuously differentiable function with $o > n$. In this situation, Newton's method begins assuming an initial guess $\mathbf{r}_0$ and then generates a sequence of iterates via

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \mathbf{J_q}^{-1}\,\mathbf{f_q}(\mathbf{r}_k). \tag{4.36}$$

In practice, this involves solving the linear system:

$$\mathbf{J_q}\,\Delta\mathbf{r} = \begin{bmatrix}\mathbf{J_{qq}} & \mathbf{J_{qp}}\end{bmatrix}\begin{bmatrix}\Delta\mathbf{q}\\ \Delta\mathbf{p}\end{bmatrix} = -\mathbf{f_q}(\mathbf{r}_k). \tag{4.37}$$

The main difference with the root-finding algorithm defined in (3.32) is that, in this case, the linear system (4.37) is under-determined, i.e., it may have an infinite number of solutions. In order to develop a well-defined algorithm, additional constraints must be imposed so that a unique step $\Delta\mathbf{r}_k$ can be defined. The most basic solution is the so-called *normal flow method*, which chooses the solution with minimum Euclidean norm. This can be easily computed by solving the QP problem:

$$\begin{aligned} &\min_{\Delta\mathbf{r}} \frac{1}{2}\,\Delta\mathbf{r}^T\Delta\mathbf{r} \\ &\text{s.t. } \mathbf{J_q}\,\Delta\mathbf{r} = -\mathbf{f_q}(\mathbf{r}_k) \\ &\qquad \mathbf{p}_m \leq \mathbf{p} \leq \mathbf{p}_M. \end{aligned} \tag{4.38}$$

In this explanation, we ignore box constraints and refer the reader to the practical considerations in section 4.2.3. The step resulting from solving (4.38) is $\Delta\mathbf{r} = -\mathbf{J_q^+}\mathbf{f_q}$, where $\mathbf{J_q^+} = \mathbf{J_q^T}(\mathbf{J_q}\mathbf{J_q^T})^{-1}$ is the well-known Moore-Penrose pseudo-inverse. This step is generally referred to as the Moore-Penrose step. This solution of the linear system is a natural choice for a Newton step because it is the shortest step from the current iterate to a root of the linear problem and, therefore, the linear model is likely to be a better representation of the nonlinear function at that step than at other solutions. Under certain assumptions –i.e., $\mathbf{f}_q$ and $\mathbf{J_q}$ continuous and differentiable and $\mathbf{J_q}$ full rank– this algorithm has proven local convergence. Fig. 4.6 (left) shows a depiction of the subsequent steps corresponding to (4.38).

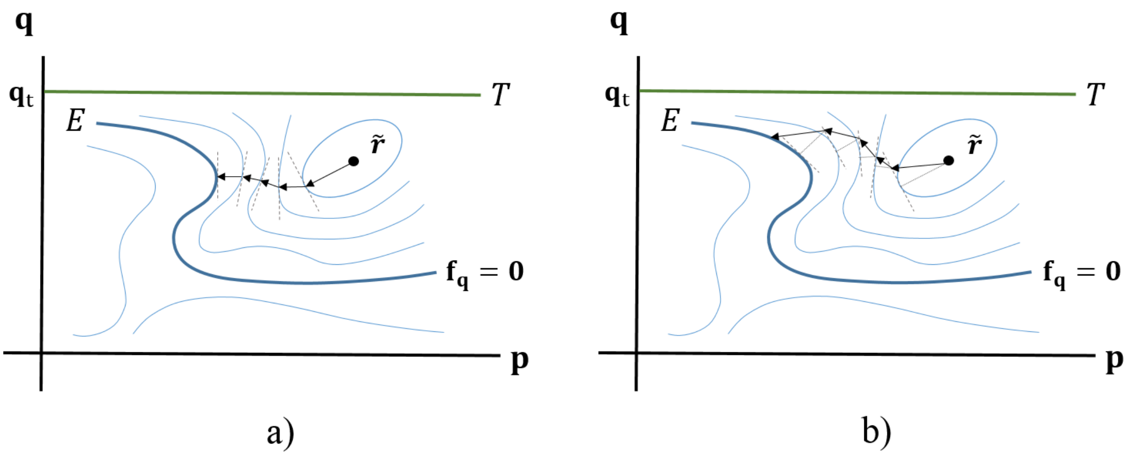


Figure 4.6: This figure shows two convergence examples for the root-finding method in total space R. The left picture corresponds to an extension of the normal flow method: the steps are necessarily aligned with the normal to the constraint defined by the linear system in equation (4.37). The right picture corresponds to a projection scheme for which the shortest step objective $\Delta\mathbf{r}^T\Delta\mathbf{r}$ used in equation (4.38) has been modified to account also for the design target.

While local convergence is guaranteed, our problems are rarely quadratic. Consequently, globalized inexact versions of the method are required. Many global extensions to the normal flow method have been proposed in the literature, implementing some kind of step length selection.

- **Line-search**. These are based on the use of a merit function, which is usually chosen to be $\Psi = ||\mathbf{f_q}||$. A line-search stage is added after the resolution of the QP subproblem in (4.38) to select α such that $||\mathbf{f_q}(\mathbf{r} + \alpha\Delta\mathbf{r})|| < ||\mathbf{f_q}(\mathbf{r})||$.
- **Trust-region**. These are based on solving the QP subproblem subject to the constraint $\Delta\mathbf{r} < \delta$, for some trust-region area δ. Characteristic algorithms of this class are *Dogleg methods*, which build a piecewise linear curve approximating the solution using a combination of the Moore-Penrose step and the point minimizing the problem (4.38) in the steepest descent direction.

For additional information on basic extensions to this algorithm, we refer the reader to [185]. Alternatively, this problem has also been studied as the minimization of a nonlinear least squares error function $f = 1/2\,||\mathbf{f_q}||^2$, using both line-search and trust-region Gauss-Newton-like formulations, for instance, the *Levenberg-Marquardt* solutions proposed in [186, 187].

Nevertheless, the normal flow method is not free of limitations:

- **Performance**. Most of the global extensions use the norm of the forces in the deformed configuration $||\mathbf{f_q}||$ as the measure of progress. However, we have seen in section 3.4 that critical points of this function could be saddle points of the potential energy V, and hence correspond to unstable equilibria. Furthermore, as introduced before, the equilibrium manifold is highly nonlinear and traversing the total space is usually full of perils. Consequently, more robust methods are required to guarantee the fast convergence of these algorithms.

- **Flexibility**. Taking always steps in the direction of the constraint normal (i.e., the Moore-Penrose direction) does not offer much improvement in terms of flexibility with respect to the potential energy minimization in deformed space Q. Ideally, we would like to control as much as possible how the algorithms converge to the nonlinear manifold, for instance, considering the distance to the design target configuration $\Delta\mathbf{t}$ within the QP subproblem formulation in (4.38) resulting:

$$\begin{aligned} \min_{\Delta\mathbf{r}} \; & \frac{1}{2}\Delta\mathbf{r}^T\Delta\mathbf{r} + \tau\Delta\mathbf{t}^T\Delta\mathbf{t} \\ \text{s.t.} \; & \mathbf{J_q}\,\Delta\mathbf{r} = -\mathbf{f_q}(\mathbf{r}_k) \\ & \mathbf{p}_m \leq \mathbf{p} \leq \mathbf{p}_M, \end{aligned} \tag{4.39}$$

for some weight τ controlling the deviation from the normal flow method. An algorithm of this kind might effectively project an approximate solution back to the equilibrium manifold, while simultaneously improving the convergence of the overall inverse design problem, Fig. 4.6 (right). However, to our knowledge, there are no theoretical studies that analyze how steps moving away from the Moore-Penrose direction might affect convergence, and further work is needed in this area.

4.3.5 Discussion

In this section, we have analyzed in detail the sensitivity-based SQP algorithm for the solution of inverse elastic design problems. We have discussed two aspects of the algorithm that critically define its performance: i) the formulation of the quadratic subproblem and ii) the projection to the constraint manifold. The optimal choice highly depends on the particular problem. During the development of this thesis, we have found that the following approaches perform significantly better than others, for the kind of design problems that we have dealt with:

- **Quadratic subproblem**. Gauss-Newton BFGS method generally outperforms other options. The BFGS update is only valid when the condition $\mathbf{y}_k^T\mathbf{s}_k > t_b$ holds; in other cases, the update yields Hessians with very large norms, and makes necessary to restart the iterative approximation, $\mathbf{B}_k = \mathbf{I}$.

We have observed that update failures happen commonly, when the full Hessian becomes indefinite or ill-conditioned, being necessary to switch to steepest descent. Gauss-Newton BFGS provides a safeguard for such situations because it switches to Gauss-Newton instead, $\mathbf{B}_k = \mathbf{S}_k^T \mathbf{S}_k$, which has better convergence properties. Moreover, Gauss-Newton BFGS provides better overall approximations to the Hessian. Nevertheless, we have seen that Gauss-Newton steps are prone to overshoot the solution, and hence a conservative step-length selection procedure is advisable. In our work, we have opted for a simple line-search and for adaptively controlling the maximum allowed step length depending on success. However, it would be interesting to consider a more complex trust-region algorithm.

- **Projection scheme**. Minimization in deformed space Q currently outperforms other options. If we restrict to the deformed space Q, the potential energy is sufficiently well-behaved, and the various safeguard techniques defined in section 3.4 guarantee the convergence and the robustness of the method. Our early attempts to employ the root-finding algorithm in total space S suggest that, in some cases, this approach does provide faster convergence rates of the inverse elastic design problem. However, projection times are often slower due to the lack of robustness.

Future work in this area arises from the limitations of our current methods. First, different higher-order approximations to the equilibrium constraint manifold could be considered. In particular, the work by Yang et al. [26] also employs second-order approximants, using quadratically parameterized osculant surfaces. It would be interesting to further study the relationship between that solution and the approaches analyzed in this chapter, seeking more accurate manifold approximations.

Additionally, the projection scheme based on the root-finding algorithm in total space R, to our knowledge, has not been explored yet in the context of inverse elastic design problem. This might open the possibility of formulating projection schemes that consider the design objective, hence improving the overall convergence. For that to be possible, we would need to formally characterize the problem, and devise techniques to overcome the numerical perils of the root-finding algorithm.

Chapter 5

Flexible rod meshes

Flexible rod meshes are lightweight structures for which heterogeneous global deformation behavior can be achieved using a single material, by locally varying the cross-section of the rods. In this chapter, we study the computational design and digital fabrication of flexible rod meshes as an implementation of deformable surfaces. Built upon the mechanical model for the simulation of rod meshes that was introduced in chapter 3, section 3.2.2, and the inverse elastic shape design framework that was studied in chapter 4, we propose an optimization-based design tool that automatically designs printable rod structures that deform with desired behavior.

- The first section 5.1 introduces the problem, motivate the use of flexible rod meshes as an option for the prototyping of deformable objects, and summarize the contributions of this work.
- The second section 5.2 describes how we have used the tools introduced in previous chapters to create a computational model of the design space and deformation behavior.
- The third section 5.3 explains how we solved the associated optimization problem using the sensitivity-based SQP algorithm described in chapter 4.
- The fourth section 5.4 presents the results and fabricated physical prototypes that validate our method and discusses limitations and potential future work.

5.1 Introduction

In this work, we explore the use of flexible rod meshes as an implementation of deformable surfaces. They are lightweight, relatively low-cost structures, that can also form the support structure of solid objects if filled or dressed. But most importantly, the global deformation properties of a rod mesh can be adjusted simply by locally varying the cross-sectional parameters of the rods. As a result, a heterogeneous deformable object can be fabricated in one piece and from a single base material using a variety of rapid prototyping technologies.

We propose a computational tool for the example-based design and fabrication of flexible rod meshes. Our method takes as input several poses of a deformable surface with known position and/or force constraints, and automatically computes a printable representation of a rod mesh that best approximates the desired shapes (Fig. 5.1). As we will see, the use of multiple deformed poses with different boundary conditions allow us to approximately define in a compact and easy manner the desired range of movements of a fabricated object. Starting from several deformation instances with different elasticity properties, we seek a unique design that generalizes all the behaviors. This facilitates reproducing complex deformation features like anisotropy, heterogeneity, or model merging (see section 5.4).

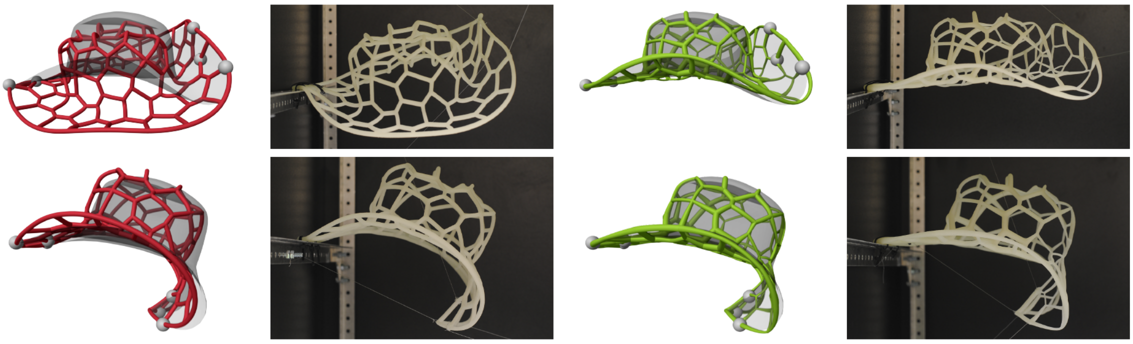


Figure 5.1: Comparison between hat simulations and printed results. The two left columns show the hat with the default rod mesh, which does not deform as desired (shown in transparent gray). The two right columns show that, by optimizing the radii and rest-shape of the rod mesh, we can fabricate in one piece a hat that deforms as desired.

One of the main features of our method is the choice of design space. We use rod meshes dominated by hexagonal faces, because hexagons can stretch, shear and bend by deforming their edges. Given such a mesh, the design space consists of its rest shape and two orthogonal radii describing the ellipsoidal cross section of the rods. By adjusting two orthogonal radii, we manage to control in-plane deformations (stretch and shear) and out-of-plane deformations (bending) of the surface independently.

In order to estimate the design parameters, we propose a simulation-based optimization approach. To this end, we extended the discrete elastic rod model [45, 46] to handle connections between multiple rods. The resulting rod mesh mechanical model has been describe in chapter 3, section 3.2.2. Built upon this simulation model, we propose an optimization framework that determines design parameters by minimizing the approximation error with respect to the input poses while satisfying static equilibrium constraints. In order to handle these constraints efficiently, we compute gradients of the objective function that satisfy the constraints implicitly using sensitivity analysis, also already introduced in the last chapter 4, section 4.3.1.

We have applied our design and fabrication method to a variety of examples. In particular, we have explored the potential of the method for toy and apparel design. In order to validate the behavior of our designs in practice, we additionally fabricated three physical prototypes.

The rest of the chapter is structured as follows. First, in section 5.2 we describe the computational model upon which this design tool is built. We briefly recall the fundamentals of the discretization used in our mechanical model and describe in depth how it is related with the definition of the design space. Then, we explain the formulation of our optimization problem and the specifics of the scheme we have used for solving it. To conclude, in section 5.4, we show some fabricated results along with a discussion on the limitations and future work.

5.2 Computational model

In this section, we describe the computational model used for representing the deformation behavior and design space of a rod mesh, following the framework introduced in chapters 3 and 4. Then, we formulate the optimization problem solved for automatically finding the design.

5.2.1 Mechanical model

As introduced in section 3.2.2, we have extended the discrete elastic rod model [45, 46] to handle connections between multiple rods. We have followed a co-rotational approach to estimate the orientations of connections kinematically from incident rods, we have defined a coupling energy and we have showed how to transmit bending and twist forces between connected rods. Our model derives completely from the classic theory of Kirchhoff rods and features elastic behavior at the connections, modeling realistic large deformation ever for relatively coarse discretizations. This fact together with the absence of numerical constraints allows us to formulate and solve efficiently a complex optimization problem.

Recall from chapter 3, section 3.2.1, that our mechanical model discretize continuous adapted-frame rod kinematics in two sets of variables: centerline nodes $\mathbf{c}_i \in \mathcal{R}^3$, which explicitly describe the configuration of the curve, and edge roll angles $\theta_i \in \mathcal{R}$, which represent the material frame as the rotation along the tangent of an adapted reference frame. Centerline nodes corresponding to rod ends incident in the same connection are shared resulting in a total of n_n nodes and n_e edges. In this work, we adopt the implicit approach for handling the rotation of connections described in section 3.2.2 and hence no additional DoF are needed. The resulting deformed configuration is $\mathbf{q} = \{\mathbf{c}, \theta\} \in \mathcal{R}^{3n_n+n_e}$ with $\mathbf{c} \in \mathcal{R}^{n_n}$ and $\theta \in \mathcal{R}^{n_e}$. We denote magnitudes corresponding to the undeformed configuration as $\bar{\mathbf{c}}$ and $\bar{\theta}$.

To ease the fabrication of the flexible rod meshes, we assume that the whole rod mesh is built with a single material, and the mass and deformation properties of the mesh can be controlled by tuning the cross-section geometry of rods. To this end, the anisotropic bending model of Bergou et al. [21] provides the appropriate flexibility. The continuous rod mesh cross-section shape is discretized at edges with a set of variables $\mathbf{r_i} = \{\mathbf{r}_i^b, \mathbf{r}_i^n\}$, with $\mathbf{r}^b$ the radius in the binormal direction and $\mathbf{r}^n$ the radius in the normal direction. Hence, for a given undeformed configuration and fabrication material properties, the deformation behavior is completely determined by $\mathbf{r} \in \mathcal{R}^{2n_e}$.

5.2.2 Design space

One of the main features of our approach is the choice of design space, which aims at minimizing fabrication complexity while maximizing design flexibility. There are two major design decisions in our approach that make this possible: i) we define a hexagonal topology and keep it constant throughout the design process and ii) we select a compact parameter set that produces a smooth design through interpolation.

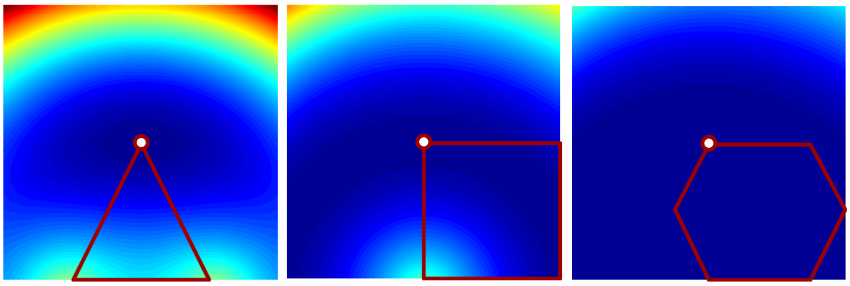


Figure 5.2: Comparison of the structural stiffness of a triangle, a quad and a hexagon made of rods. We fixed the bottom-most edge of the structure, enforced the indicated vertex to be at each position in the field, and computed the static equilibrium. The plot shows the resulting deformation energy. Notice that the area of low deformation energy under stretch is clearly bigger for the hexagonal mesh.

Hexagonal topology

We seek for a mesh topology that minimizes structural stiffness, thus enabling a larger range of feasible designs under the same magnitude of external forces. Rods are radically more compliant in bending than stretch, hence the topology of the mesh should be one that enables deformations in all directions simply by bending rods as seen in Fig. 5.2. We have opted for a mesh topology dominated by hexagonal faces, as hexagons can deform in all directions by bending their edges. Fig. 5.3 compares the structural stiffness of a triangle-mesh, a quad-mesh, and a hexagon-mesh of rods for a particular deformation scenario. The hexagon-mesh can stretch and shear in all directions, the quad-mesh resists stretch in directions aligned with quad edges, and the triangle-mesh is almost inextensible.

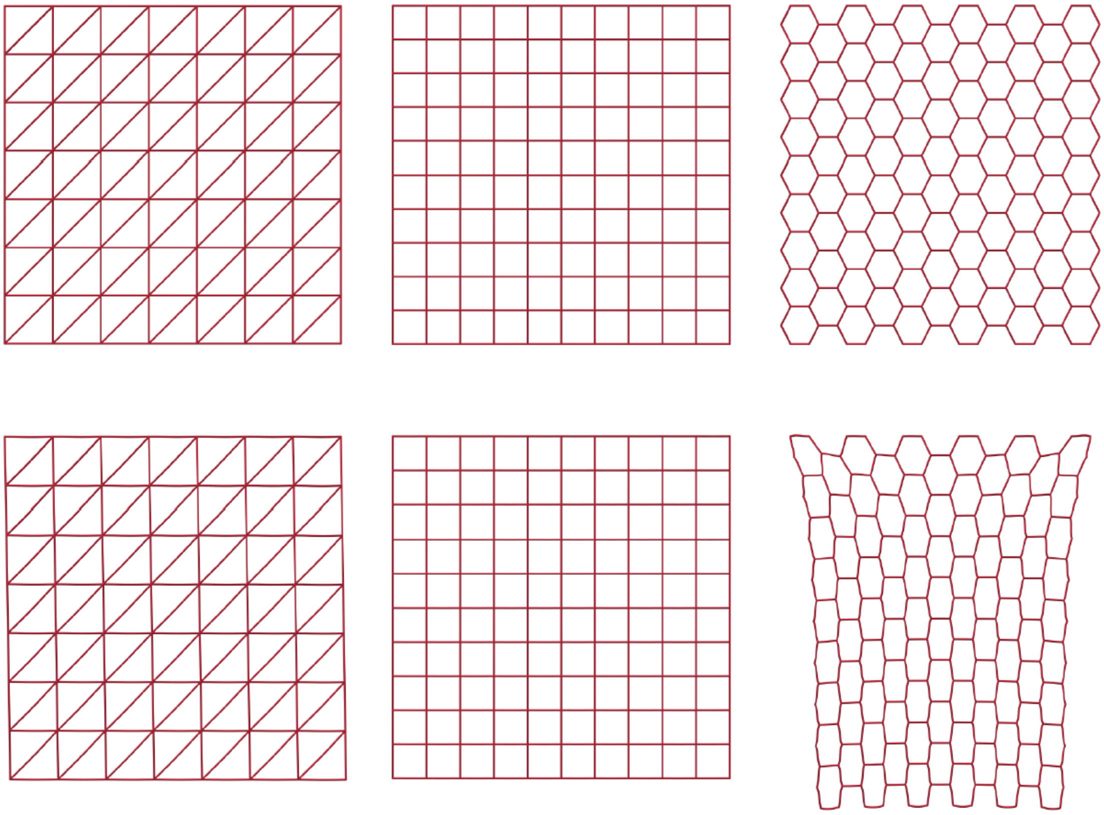


Figure 5.3: With the same material and mesh density, a hexagonal mesh is more compliant to stretch and shear deformations than a triangle or quad mesh, thereby increasing the flexibility of our optimization method. In this example, we show meshes with approximately the same number of cells, in their rest configuration (top), and after hanging from them the same weight (bottom).

To construct a mostly hexagonal mesh, we have explored two options in our examples. One is building the dual mesh of an input triangle mesh. The other one is to initialize a number of samples on a surface and then compute a *centroidal Voronoi tessellation* [188]. With hexagon meshes, we have observed that our optimization method is largely insensitive to the specific mesh topology. Fig. 5.4 shows an optimization result that is matched using three different meshes.

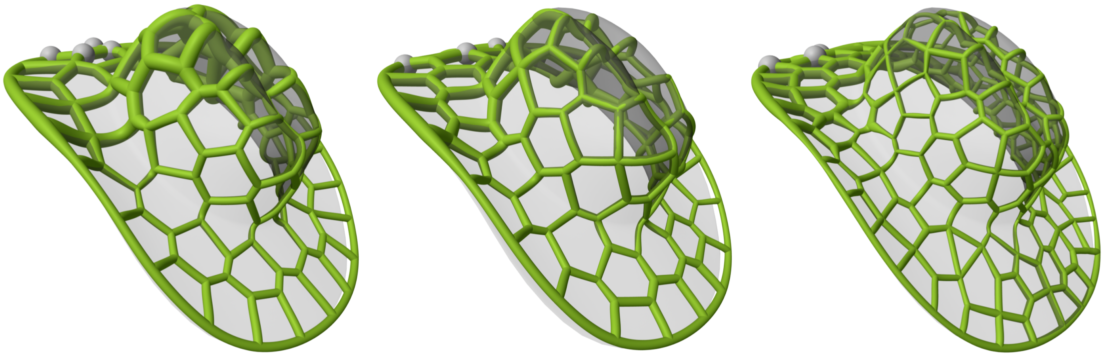


Figure 5.4: A hat mesh is optimized to reach the target deformation (shown in light gray) with three different meshings. All meshes were obtained through Voronoi tessellations; the first two with 64 cells and different initializations, and the third one with 128 cells.

Compact parameterization

Our design space consists of the rest shape of the rod mesh $\bar{\mathbf{c}} \in \mathcal{R}^{3n_n}$ and the two radii of the elliptical cross-section at each point of rod mesh $\mathbf{r} = \mathcal{R}^{2n_e}$. By adjusting the radii, we manage to control the overall stiffness of the mesh. Tuning also the rest configuration allow us to obtain even more accurate results when considering multiple target poses.

Both parameter sets change the geometry of the mesh and hence the appearance of the printed object. Wiggly designs affect the aesthetics of the mesh but also compromise the capability of the discretization to approximate the continuum deformation behavior. Furthermore, in this multi-objective scenario, there is high chance of overfitting. This problem not only reduces the generalization properties we seek, but also produces noisy results if the optimization method locally overfits for one particular target. With these observations in mind, we follow the heuristic for the definition of a compact design space with naturally smooth shapes introduced section 4.2.1.

Both for radii and rest positions, we set as design parameters control points along the rods $\mathbf{c}_c$ and $\mathbf{r}_c$. We place four control points per rod, two at the end connections and two evenly spaced, and smooth their values using cubic Hermite interpolation. Overall this choice results in a parameter vector $\mathbf{p} = (\mathbf{r}_c, \bar{\mathbf{c}}_c)$, with $(\mathbf{r}, \bar{\mathbf{c}}) = \mathbf{Z}\,\mathbf{p}$, for a precomputed matrix $\mathbf{Z}$ of cubic Hermite coefficients. To further improve the visual appearance and deformation behavior of the resulting rod meshes, we also enforce continuity and alignment of cross-sections at connections in two ways:

1. **Connection sharing**. Analogously to the coupling technique used for mechanical simulation, we share control points among all rods incident in a connection, such that radii magnitude smoothly vary throughout the rod mesh.

2. **Normal alignment**. We keep material and reference frames at connections aligned in the rest configuration. Whenever the rest shape changes, we fit a plane to the incident edges of each connection. We use the normal of this plane as the normal axis of rest-shape frames of incident edges. This choice typically defines the first and the last rest frames of each rod in the mesh; the alignment of the other frames of the rod is linearly interpolated from the extremal values. We span the orthogonal cross-section radii at each point in the rod mesh using material frames. Thus, as an additional effect of frame alignment, we gain more flexibility during optimization because adjusting the two orthogonal radii is guaranteed to control tangent plane and out-of-plane deformations.

5.3 Optimization problem

Our optimization framework takes as input a few deformed poses of a rod mesh under known boundary conditions. Then, it automatically computes the rest shape and cross-section of the rods such that the mesh best matches the input poses under the same boundary conditions. The resulting rod mesh geometry is finally used as input for an automatic fabrication process.

In this section, we first describe the general optimization framework. We use a Newton optimization method subject to boundary conditions, design constraints, and static equilibrium constraints, which are enforced implicitly. Then, we describe an optimization scheme to optimize both the material (i.e., radii) and rest shape of the rod mesh.

5.3.1 Numerical optimization

Following the notation described in chapter 4, our optimization framework receives as input a set of k_p target poses $\{\mathbf{q}_k^t\}$ in static equilibrium, and for each pose a subset of DoFs $\mathbf{b}_k = \mathbf{B}_k\,\bar{\mathbf{q}}_k$ specified as fixed boundary conditions. Here $\mathbf{B}_k$ is a per-pose selection matrix. The method also supports using forces as boundary conditions.

We denote as $\mathbf{p}$ a generic set of m design parameters, which could be shape parameters (i.e., rest-shape coordinates of rod nodes) and/or material parameters (i.e., radii of rod cross-sections). We then formulate the objective as the squared error between rod node positions and the input poses:

$$h(\mathbf{p}, \{\mathbf{q}_k^t\}, \{\mathbf{b}_k\}) = \frac{1}{2}\sum_{k=1}^{k_p} w_k \|\mathbf{\Phi}(\mathbf{q}_k) - \mathbf{\Phi}(\mathbf{q}_k^t)\|^2, \tag{5.1}$$

where $\mathbf{\Phi}$ is a shape descriptor that selects only the nodes of the deformed and target configuration. Then, we pose a constrained optimization that minimizes this objective function, subject to static equilibrium, box constraints on design parameters, and the input boundary conditions:

$$\begin{aligned} &(\mathbf{q}, \mathbf{p}) = \arg\min h(\mathbf{p}, \{\mathbf{q}_k^t\}, \{\mathbf{b}_k\}), \\ &\text{s.t. } \mathbf{f}_{\mathbf{q},k}(\mathbf{q}_k, \mathbf{p}, \mathbf{b}_k) = \mathbf{0}, \forall k. \\ &\mathbf{p}_m \leq \mathbf{p} \leq \mathbf{p}_M \end{aligned} \tag{5.2}$$

$\mathbf{f}_{\mathbf{q},k}$ denotes a vector that concatenates the forces on all rod nodes for the k^{th} pose, w_k is a scalar to weight target poses, and $\mathbf{p}_m$, $\mathbf{p}_M$ are minimum and maximum constraints on design parameters. To solve this constrained optimization problem, we follow the sensitivity-based SQP approach defined in section 4.3.2: we enforce the equilibrium constraints implicitly and iterate QP updates of the design parameters with box-constraint projection.

In practice, for the calculation of the sensitivity matrix $\mathbf{S}_k$ defined in section 4.3.1, we compute $\mathbf{J}_{\mathbf{qq},k}$ analytically, but we evaluate $\mathbf{J}_{\mathbf{qp},k}$ using finite differences to facilitate the exploration of arbitrary design parameters. By handling the static equilibrium constraints implicitly, the minimization in (5.2) turns into a box-constrained nonlinear least-squares problem. To handle box constraints on the radii, we simply project the gradient for those parameters that have reached a limit. Depending on the approximation of the Hessian of (5.2) and the equilibrium projection scheme, the convergence properties of the algorithm vary.

- **Quadratic subproblem**. Computing the full Hessian of (5.2) is prohibitive, so we have tested approximate methods instead, specifically Gauss-Newton with line-search and Levenberg-Marquardt. Both methods approximate the Hessian using the sensitivity matrix, and we found them to behave similarly in our examples. We have also observed that the full Hessian occasionally becomes indefinite and its approximations ill-posed. This induces close to non-descendent search directions, being necessary to switch to steepest descent. As a compromise solution, BFGS with line-search provides enough robustness as Hessian updates are often invalid in such situations, thus switching to steepest descent. Plus, controlling the step length is key to the performance of our method; therefore, we enforce a maximum length which is adaptively refined depending on success.

- **Equilibrium projection**. After every QP update of the parameter vector $\mathbf{p}$, we solve a static equilibrium problem on all poses. To do this, we use the energy minimization solver introduced in section 3.4.

Finally, to apply position boundary conditions $\{\mathbf{b}_k\}$, we found that fixing simulation nodes increases excessively the error on free nodes. Instead, we enforce boundary conditions using a penalty force, and we progressively adjust the penalty stiffness such that the error at constrained nodes is similar to elsewhere in the mesh.

5.3.2 Optimization scheme

As already mentioned, the vector of design parameters $\mathbf{p}$ may be formed by material parameters (i.e., rod cross-section radii) and/or shape parameters (i.e., rest-shape coordinates of rod nodes). We have observed that adjusting cross-section radii has the largest effect on the bulk fitting error, while adjusting rest-shape coordinates increases fitting quality for designs that are close to the optimum and have to deal with objectives. For this reason, we propose the following optimization scheme.

We start with a multiresolution optimization of the material of the rod mesh. We first optimize only the radii control points at connections, linearly interpolated along rods; then we add radii control points in the middle of rods, with quadratic interpolation; and we finally optimize for all radii control points with cubic interpolation. Once the material optimization alone has converged, we start iterating steps of material and shape optimization directly on all control points. Please see Fig. 5.5 for examples of convergence with our optimization scheme

5.4 Experimental results

In all our examples we follow a similar design pipeline, with small variations. We start with a surface description of the deformable object to be fabricated, typically a high-res triangle mesh $\mathcal{H}$. At this point, we define the connectivity of the rod mesh $\mathcal{R}$ to be optimized. If the user does not provide a specific topology, we automatically compute one such that it is close to an hexagonal mesh. We project the resulting connections onto $\mathcal{H}$, and construct rods connecting them by following geodesics. For each vertex of $\mathcal{R}$, we store a mapping to its projection triangle in $\mathcal{H}$.

To create the target poses, we apply some deformer to $\mathcal{H}$. In our examples, we have explored different deformers, mainly direct artist manipulation and embedding in another physical model. Given a target deformation defined on $\mathcal{H}$, we can determine the target configuration of the rod mesh by its stored mapping.

Table 5.1 summarizes the rod mesh size, complexity, continuum material parameters and radii constraints for all our benchmarks. We have printed some of the obtained results to serve as a validation of both the simulation model and the optimization procedure. In all the cases, we have used laser sintering technology with the material TPU 92A (elastic modulus $E = 32$MPa, Poisson ration $\mu = 0.48$ and density $\rho = 1200$Kg/m^3).

Model	Sheet	Hat	Dino	Smiley
Nodes	693	1727	3219	2663
Edges	710	1790	3330	2892
Rods	71	179	333	723
Connections	54	116	222	494
Material Params.	392	948	1776	2664
Rest-Shape Params.	588	1422	2664	3663
Young mod. (MPa)	32	32	32	10.0
Poisson ratio	0.48	0.48	0.48	0.48
Density (Kg/m^3)	1200.0	1200.0	1200.0	1000.0
Size (m)	0.2	0.23	0.20	0.25
Max. Radius (mm)	5	3	3	4
Min. Radius (mm)	1	1	1	1
RMS Error (mm)	1.03	1.13	0.56	0.41
Computation Time	35min	1h45min	2h20min	3h10min

Table 5.1: Statistics of the benchmarks: model discretization, optimization complexity, mechanical parameters, design constraints, resulting RMS error, and computation time.

5.4.1 Performance

Our rod structure model introduces additional computational complexity to the discrete elastic rods approach. Computing the energy gradient and Hessian at connections is costly and involves a reduction in performance. However, we are aiming for sparse rod structures therefore we can expect the overall complexity to grow linearly on the number of rods.

Our optimization scheme considers implicit constraints for each target pose, which are independent from each other. The step is broken into a set of individual problems which can be solved separately, and the complexity grows linearly in the number of target poses. As a counterpart, two operations are specially costly: i)

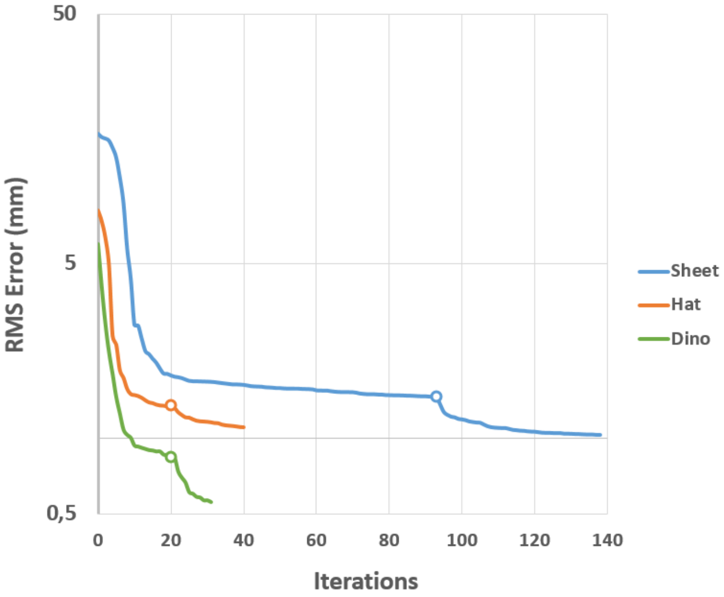


Figure 5.5: Convergence of the optimization for the three printed results, showing the evolution of RMS error (in log scale). White dots indicate the iteration from which the optimization alternates radii and rest-shape optimization.

performing the sensibility analysis and ii) computing the static equilibrium.

The cost of the sensitivity analysis is partially alleviated by incrementally adding resolution to the parameterization. As shown in Fig. 5.5, error is greatly reduced in the first steps of the optimization, when only one control point per connection is used. However, improvement gained by subsequentially refining the parameterization and iterating radii and rest pose optimization steps is significant. At this stage of the optimization, steps are more time-consuming so a trade-off must be considered between computation time and the quality of the solution.

The cost of computing the static equilibrium is highly dependent on the step length, as the static solve is computed from the previous equilibrium configuration. If the new static equilibrium is too far from the previous one, it is more convenient to compute equilibrium from the rest configuration. Moreover, convergence problems may activate line-search bisections, leading to additional static solves. Adaptively controlling the maximum allowed step length, along with using the quasi-Newton update helps to reduce failed steps and has an important positive impact on performance. Table 5.1 shows an approximation of the time needed to converge to the optimal solution for each benchmark.

5.4.2 Printed examples

Sheet

Our first example consists of a simple sheet (Fig. 5.6), showing the capability of hexagonal rod meshes of exhibiting varied behaviors. The rod mesh is 0.20m side and consists of 71 rods and 54 connections forming a regular hexagonal pattern, with a total of 693 nodes.

Four target poses are defined by computing the static equilibrium of the rod mesh for different radii configurations and boundary conditions (fixed points and weights). These targets are designed to be opposing: the first pair (first and second row) imposes bending anisotropy; the second pair (third and forth row) imposes stretching capabilities. The optimized rod mesh effectively captures all required behaviors. Simulated results has been validated by testing physical printed realizations, which closely match the behavior predicted by the simulation. Although low frequency deformations are perfectly replicated, small differences appear which may be cause by inaccuracy in the validation process.

The images in Fig. 5.6 show in semi-transparent gray the target configuration of the rod mesh. The default behavior at the beginning of the optimization is shown in red against the deformations obtained with our optimization framework in green. Cross-sectional radii optimization is capable of reducing the RMS error to 1.4756mm per node. We achieve an overall fitting of 1.0322mm upon convergence of rest-shape optimization.

Hat

Our second example is a hat of approximately 0.23m long, shown in Fig. 5.7 and Fig. 5.1. The connectivity of the rod mesh is automatically computed using a centroidal Voronoi tesselation on a high-res triangle mesh. This yiedls to a close-to-hexagonal mesh consisting of 179 rods, 116 connections and a total of 1727 nodes.

Five target poses (rows in Fig. 5.7) are defined using a thin-shell deformer on the triangle mesh. We have designed several heterogeneous models and used them to create target poses. Target poses 1, 2 and 3, use a model where the front side of the hat is much softer than the back side. Target pose 4 uses a homogeneous model and is intended to make the hat maintain its rest shape under gravity. Target pose 5 uses a model where the left side of the hat is clearly softer than the right side. We fix some points and pull from handles using forces to achieve interesting deformations. The deformed position of the handle is considered a boundary condition for our simulated rod mesh.

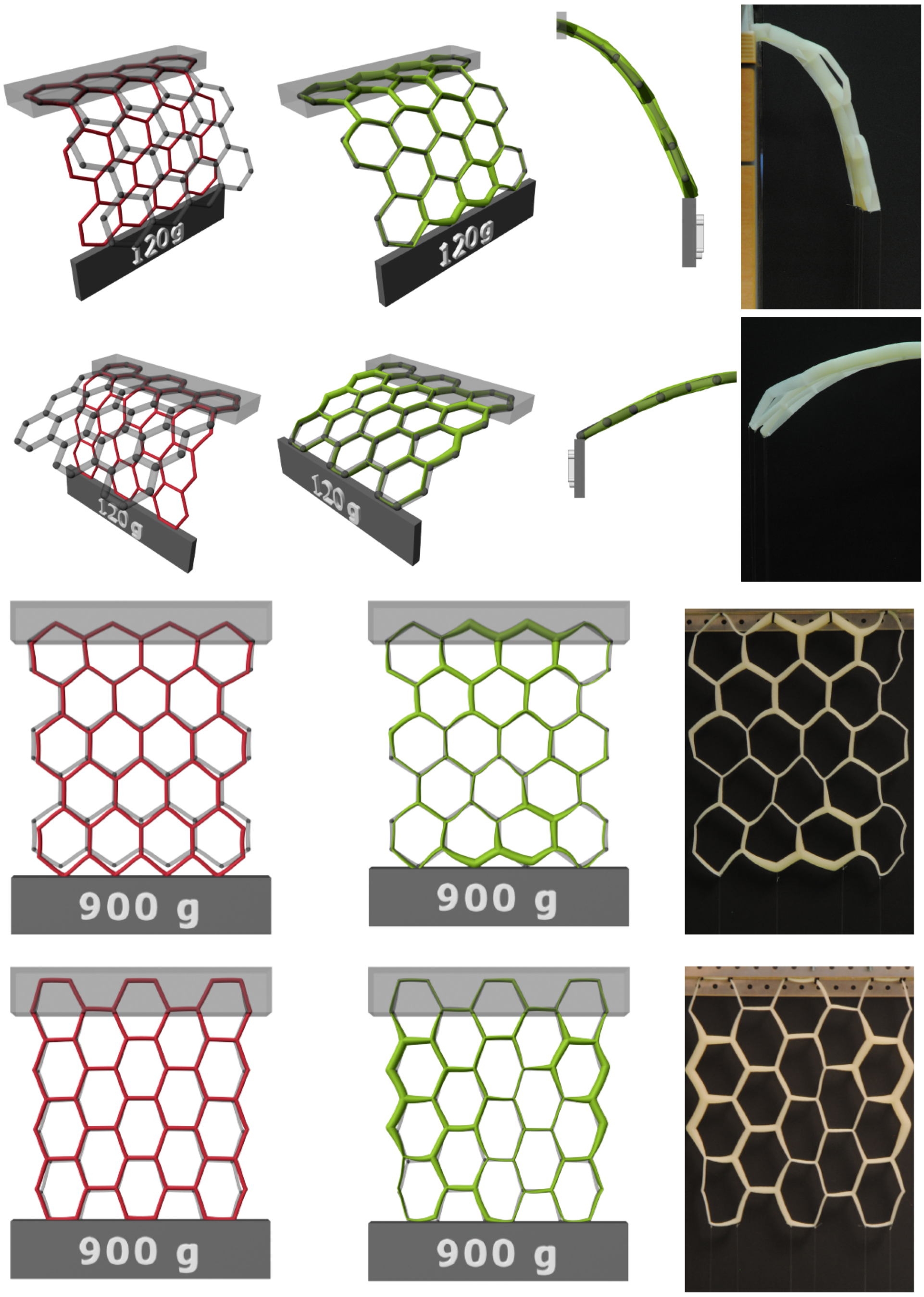


Figure 5.6: Resulting deformations for each of the four target poses of the sheet demo (shown in light grey). Default deformations (in red) are compared against the optimized results obtained using our framework (in green). The last column shows the physical printed sheet matching the behavior predicted by the simulation model. Target configurations of the rod mesh are shown in light transparent gray.

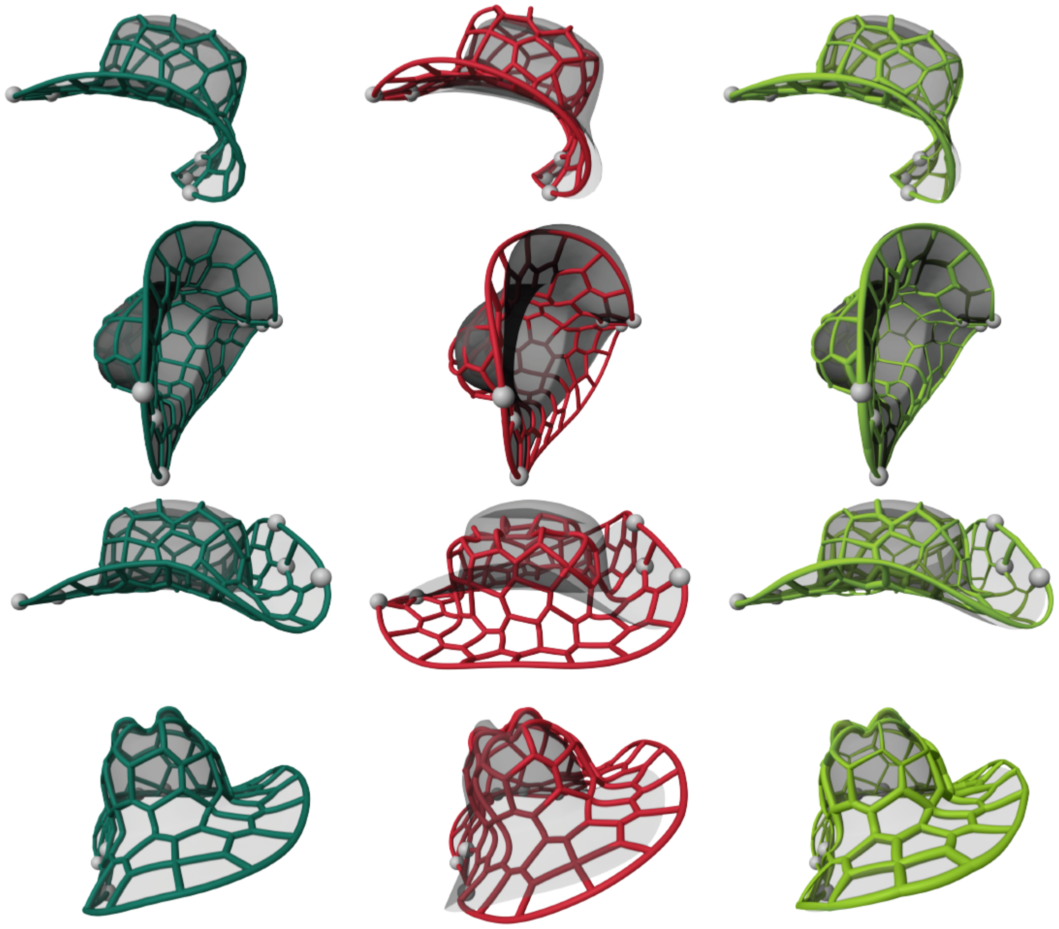


Figure 5.7: Resulting deformations for four of the target poses of the hat demo (shown in blue). Default deformations (in red) are compared against the results obtained using optimization (in green). An overlay in transparent gray is added to help understand how far the solution is from the original target deformation.

As shown in Fig. 5.7, our optimization framework is capable of combining all behaviors into a single hat. The first column shows in blue the target deformation of the rod mesh, according to the thin-shell model. The second column shows in red the deformation of the rod mesh with default parameters. The third column shows in green the final optimized mesh given by our method. We achieve an overall fitting RMS error of 1.3482mm per node, solely by optimizing cross-section radii. In this case, considering rest-shape does not improve the solution dramatically, reducing it down to 1.1128mm.

The images in Fig. 5.1 provide a visual validation of the results, for two of the target poses considered. Both default and optimized configurations of the rod mesh have been printed and fixed to a supporting structure. We tie wires to the boundary points in our simulated rod mesh and pull from them until approximately reaching their deformed position. The resulting deformation is visually appealing and resembles our model prediction for both default and optimal configurations.

Dinosaur

This example consists of a dinosaur toy of approximately 0.21m long, shown in Fig. 5.8. The connectivity of the rod mesh is semi-automatically computed from a high-res triangle mesh. We apply mesh decimation to obtain a low-res version $\mathcal{L}$, in the order of hundreds of triangles, and construct its dual mesh. That is, we place a rod connection at the centroid of each triangle in $\mathcal{L}$ and project it into $\mathcal{H}$. Then, we set a rod between two connections if their corresponding triangles in $\mathcal{L}$ share an edge. This yields to a mesh consisting of 333 rods, 222 connections, and 3219 nodes.

Five target poses are defined using an FEM deformer on $\mathcal{H}$. We have designed a heterogeneous FEM model, where the tail and the joints between the legs and the body are notably softer. The head is heavier and it should tip the dinosaur under gravity. To set boundary conditions during posing, we select a handle on the mesh and pull using forces. Using forces instead of translating the handle keeps the deformations plausible and easier to reproduce with real-world materials.

The images in Fig. 5.8 show both simulated and real results. Semi-transparent grey represents the target pose, as defined by the high-res mesh $\mathcal{H}$ deformed using FEM. The first column shows in red the behavior of the rod mesh for the default parameters. The second column shows in green the performance of our method. The third column in Fig. 5.8 shows the resulting deformation of the real printed dinosaur. For this demo, we follow the same validation methodology described above for the hat, obtaining similar results. Cross-section radii optimization achieves an already low fitting RMS error of 0.8622mm. Alternating rest-pose and full resolution material optimization is capable of further improving the solution to 0.5564mm.

Smiley

Our last result (Fig. 5.9) is intended to demonstrate the performance of our method for an arbitrary user-defined target pose. For this demo, we apply a centroidal Voronoi tessellation to a high-res triangle mesh of a sheet of 0.25m. We use a serious smiley texture as a biasing weight for face areas. This creates a hexagonal mesh where cell distribution is determined by the color of the texture. In particular, the density is incrementally higher at the face, mouth and eyes of the figure. The resulting rod mesh is formed by 723 rods, 494 connections and a total of 2663 nodes.

The single target is manually defined by an artist, geometrically deforming the

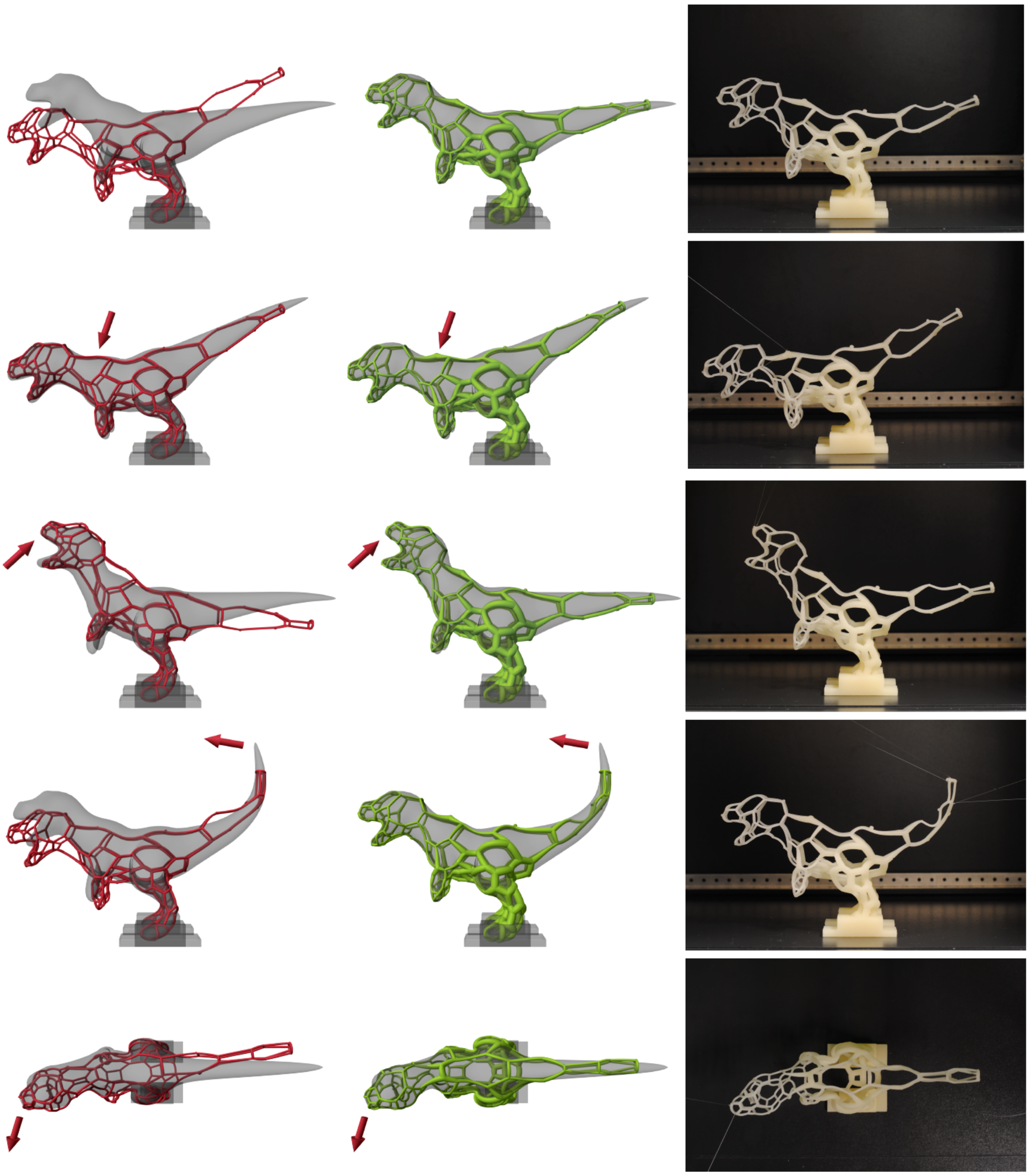


Figure 5.8: Resulting deformations for each of the five target poses of the dinosaur demo. Default deformations (in red) are compared against optimized results (in green). An overlay in transparent gray represents the target surface generated by the FEM deformer. Right column shows our visual validation of obtained results done by wire pulling from position boundary conditions.

original texture. We map each point in the rod mesh with a texture coordinate and apply the same displacement to create the target mesh pose. Our intention is to make the character smile when stretched in one direction; this simple setup entails

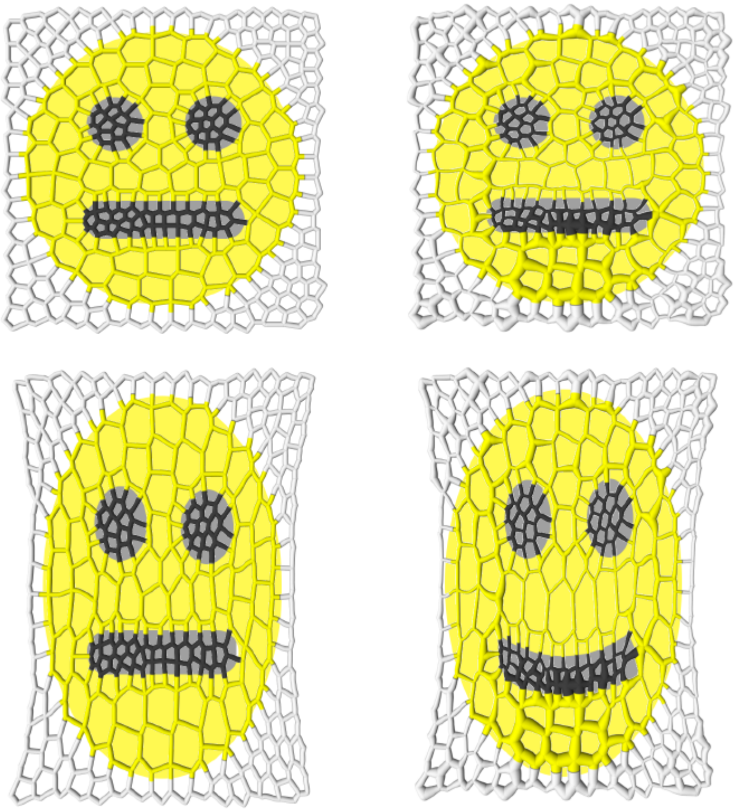


Figure 5.9: This example shows a planar rod mesh that, when stretched, produces a smiley. In this example we combine the use of a mesh with an aesthetic design (i.e., higher rod density in the mouth and eyes) and an artist-painted target (i.e., the smiley drawn behind the meshes). On the left, we show the default mesh. When stretched, its deformation is nearly uniform. On the right, we show the optimized rod mesh. When stretched, it produces a smiley. Our optimization method automatically places thick rods under the mouth and thin rods over it.

two main difficulties. First, for a homogeneous radii throughout the mesh, variable rod densities tend to create heterogeneous deformations that might not match the target texture. Second, artist deformations are not based on any physical model and hence the target pose might not be reachable at all by a mesh deformed using simple stretch forces.

The images in Fig. 5.9 compare default and optimized deformations to the warped texture. An overlay with the typical smiley colors is added to help understand how the texture has been deformed in each case. Our optimization framework is capable of closely matching the smile with almost no rest-shape deformation close to the mouth. Moreover, it approximates the overall shape of the deformed texture much better compared to the default configuration. We achieve an overall fitting RMS error of 0.4854mm.

5.4.3 Discussion and future work

We have described a computational design method that allows fabricating flexible objects with a desired deformation behavior. We opt to represent such flexible objects with meshes of deformable rods. Using a mechanical simulation algorithm, we have designed an optimization scheme that allows us to estimate the material and rest-shape of rods such that the object approximates a set of input poses.

Future work includes extending the computational design and optimization of rod meshes to the general problem of finding the correct representation for a desired deformable object. In this work, we have seen the importance of selecting an adequate topology for the structure, one which does not impose such a strong constraint on in-plane stretch. In that sense, the design method could be further extended as well. The quality of the results could be improved by incorporating topology optimization, but this may require a new definition of the objective function to appropriately evaluate surface deviation under arbitrary rod sampling.

A possible limitation of this particular representation is the possibility of finding local minima. Our optimization scheme may suffer from this problem as we alternatively optimize for material and rest-shape, and convergence is not always guaranteed. We try to address this problem through multi-scale optimization, i.e. we start from a coarser representation and progressively subdivide rods when needed. Yet, there might be some other way to avoid this problem.

Finally, an important goal is to provide the final user with the necessary tools to interactively design this kind of representations. Both simulation and optimization would be needed to convey a design space of physically plausible solutions which a designer could further refine. We will explore this possibility in the next chapter 6, in the context of Kirchhoff-Plateau surfaces.

Chapter 6

Kirchhoff-Plateau surfaces

Kirchhoff-Plateau surfaces (KPS) are planar rod meshes embedded in tense fabric that deploy into complex 3D shapes. In this chapter, we study the computational design and automatic fabrication of these structures. We create a mechanical model for the simulation of tensile structures coupling the thin-shells and rod mesh models described in chapter 3 using collocation. Built upon this mechanical model, we propose a user-guided but computer-assisted tool that allows to interactively design such structures, also considering the inverse problem studied in chapter 4.

- The first section 6.1 introduces the concept of Kirchhoff-Plateau surfaces, motivate the use of these structures, and provide an overview of the contributions of this work.

- The second section 6.2 analyzes in detail the kinematics and deformation properties of KPS, and justify the need for an interactive computer-assisted solution.

- The third section 6.3 explains how we used the computational tools described in previous chapters to build models of the design space and deformation behavior of KPS.

- The fourth section 6.4 describes in detail the different editing operations that have been implemented to facilitate the design of KPS, focusing on the solution of the inverse design problem.

- The fifth section 6.5 presents the results and fabricated physical prototypes that validate our method, and discuss limitations and potential future work.

6.1 Introduction

Minimal surfaces have intrigued scientists and engineers for more than 250 years. Their origins trace back to a problem originally raised by Lagrange that later became known as the *Plateau problem*: finding a surface of minimum area that spans a given rigid boundary curve as shown in Fig. 6.1. From an application point of view, minimal surfaces are appreciated not only for their smooth aesthetic appearance, but also for their inherent material efficiency and structural stability. For these reasons, minimal surfaces are widely used for light-weight and cost-efficient structures, ranging from large-scale roofs, canopies, and shade systems, to acoustic deflectors, light diffusers, and decorative elements for interior design.

One common way of bringing designs of minimal surfaces to practice is by using fiberglass or metal rods embedded in stretched fabric, with the added benefit that all components are planar and easy to manufacture. However, the advantages in terms of weight, cost, and fabrication come at the price of a much more difficult design problem—the *Generalized Plateau problem* of finding a minimal surface whose tensile forces are in equilibrium with the bend and twist forces of a given elastic rod [133, 189].

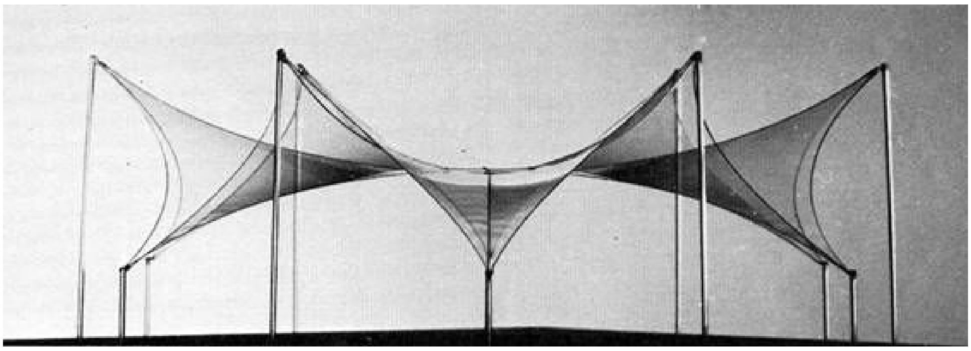


Figure 6.1: Traditional works on architectural tensile structures studied the so-called Plateau problem experimentally by considering the surface spanned by soap film embedded in rigid wires. Photograph credits to the famous architect Frei Otto.

In this work, we explore the design and fabrication of *Kirchhoff-Plateau surfaces*, i.e., networks of thin elastic rods embedded in pre-stretched membranes. We focus on surfaces that can be manufactured by 3D-printing planar rods onto stretched fabric—a process that was beautifully demonstrated in recent work by [35]. Designing in this space gives rise to two main challenges.

- First, the path to stable surfaces is fraught with perils such as nonlinearities, unstable equilibrium points, and bifurcations that lead to multiple stable configurations.

- Second, the space of possible designs is restricted by the fact that (i) KPS consist of minimal surface patches that, inherently, can only assume shapes with vanishing mean curvature, and (ii) the boundaries of the minimal surface patches can only assume shapes corresponding to equilibrium states of planar rods coupled to a stretched membrane.

In light of these challenges, we cannot expect that there exists a KPS that closely approximates a given target shape in the general case. But even when precise shape approximation is not possible, the space of KPS nevertheless provides ample room for shape abstraction and interpretation. We therefore turn away from fully-automated solutions in favor of a user-guided but computer-assisted design paradigm.

We propose a fabrication-oriented design tool for KPS –planar rod networks embedded in pre-stretched fabric that deploy into complex, three-dimensional shapes. Similar in spirit to mesh-based modeling tools, the user is responsible for creating the topology of the rod network and for transforming it into the desired shape. During this process, the user can draw from a set of modeling tools that implement simple editing operations directly on the equilibrium state of the surface. Visualization tools that indicate possible changes conforming to the editing goals further assist the user in making informed decisions. Once finished, the design can be easily manufactured using any consumer-level FDM 3D printing device, (Fig. 6.2).

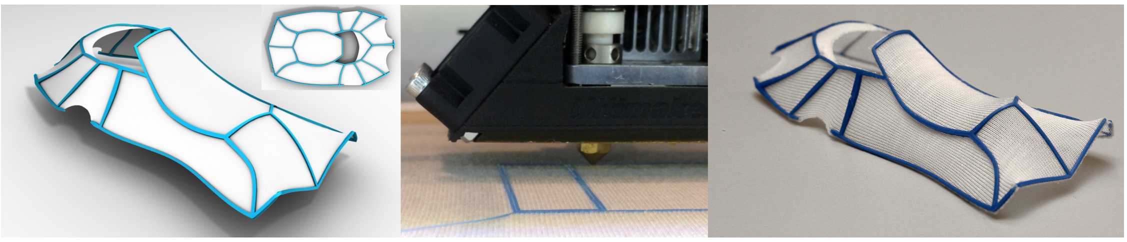
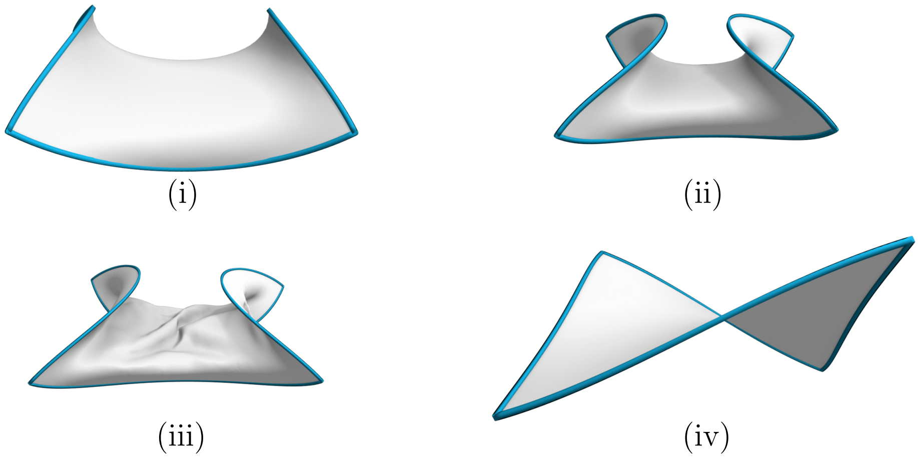


Figure 6.2: Our simple fabrication method allows to rapidly manufacture KPS structures by directly printing the designed rod mesh on top of pre-stretched fabric using any consumer-level FDM 3D printer.

This seemingly simple design approach is enabled by simulation and optimization algorithms that translate editing operations into corresponding parameter updates, and analyze the current structure in order to provide feedback on the space of feasible edits. Our simulation and optimization algorithms involve several novel features. We introduce tools for the exploration of the design space through eigenanalysis of the sensitivity matrix subject to design goals. Owing to the complex and constrained space of KPS, we furthermore cast inverse design operations as a two-step optimization process: first, in order to interactively explore first-order feasible target shapes, we combine efficient constraint projection with sensitivity-based linearization of equilibrium constraints; second, given a first-order feasible target shape, we perform nonlinear constrained optimization to compute fully-feasible designs. We demonstrate our method by designing a diverse set of complex-shaped KPS, each validated by physically-fabricated prototypes.



(i)

(ii)

(iii)

(iv)

Figure 6.3: The design space of KPS is highly nonlinear and the effects of parameter changes are often difficult to predict. The images show a simple square frame (120 × 120 mm) with different fabric pre-stretch factors s, normal rod radius r_n (governing out-of-plane resistance), and binormal rod radius r_b (governing in-plane resistance). From left to right: (i) ($s = 1.3$, $r_n = 0.2$ mm, $r_b = 0.4$ mm) base shape; (ii) ($s = 1.6$, $r_n = 0.2$ mm, $r_b = 0.4$ mm) higher stretch increases curvature; (iii) ($s = 1.3$, $r_n = 0.2$ mm, $r_b = 0.2$ mm) lower in-plane resistance leads to sagging and wrinkling; (iv) ($s = 1.3$, $r_n = 0.4$ mm, $r_b =$ 0.4 mm) higher out-of-plane resistance leads to a completely different stable configuration.

6.2 KPS shape space

Kirchhoff-Plateau Surfaces are flexible structures made from networks of elastic rods embedded in pre-stretched textile membranes. Although fabricated in a planar state, KPS can deploy into complex three-dimensional shapes that are governed by the balance between membrane and rod forces. These equilibrium shapes are influenced by a number of factors pertaining to the rod network and the membrane. In addition to the material of the rods, their resistance to bending and twisting is determined by their cross-sectional geometry. By varying this geometry, it is possible to control the ratio between in-plane and out-of-plane bending stiffness, as we have seen in the previous chapter 5. Membrane stretch, on the other hand, induces compressions in the rods, leading to unstable, planar equilibrium configurations that resolve into bending and twisting upon slight perturbations. As shown in Fig. 6.3, the ratio between membrane stretch and out-of-plane bending resistance is, effectively, a means of controlling the amount of curvature in the equilibrium shape. However, a sufficient amount of membrane stretch is also required to ensure that the surface remains tense and free of sagging or wrinkling; see Fig. 6.3-iii.

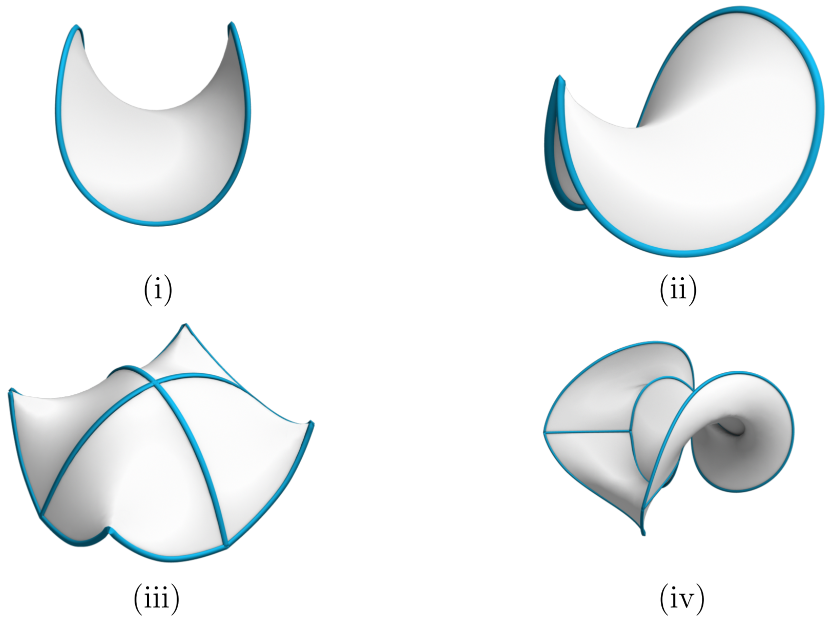


Figure 6.4: Disks of the same size and pre-stretch produce diverse shapes depending on their internal topology. From left to right: (i) empty interior; (ii) split rod on the boundary; (iii) two crossing rods in the interior; (iv) an inner disk connected to the outer boundary.

From a designer's perspective, the mechanics of KPS alone provide no direct insight into the space of shapes that can be achieved. However, even though fabric membranes are not strictly area minimizing, the intuition about KPS can be strengthened by considering them as *piece-wise minimal surfaces*: the rod network induces a decomposition of the surface into membrane patches, each bounded by a closed loop of rods. As a minimal surface, the area gradient vanishes everywhere inside each patch, which is equivalent to vanishing mean curvature. Since planar KPS configurations are generally unstable, principal curvatures have the same nonzero magnitude but opposite sign, leading to strictly negative Gaussian curvature everywhere inside a patch. Although the constraint on strict equality of principal curvatures is somewhat mitigated by real-world textile membranes, the sign constraint still applies: membrane patches can only assume so called *anticlastic* shapes with negative Gaussian curvature in every point. Moreover, the fabrication constraint that KPS must have a planar rest state implies that the 3D rod network must be embeddable in 2D without compressions, further restricting the space of possible surfaces. Nevertheless, while a single KPS patch is necessarily anticlastic, rods introduce discontinuities in the surface normals of adjacent membrane patches.

Thanks to this property, it is possible to approximate surfaces with overall positive Gaussian curvature by connecting anticlastic patches. Fig. 6.4 shows examples of diverse shapes that can be obtained by varying the internal mesh topology of a disk.

In summary, the space of KPS offers interesting and complex 3D shapes that can be created using a simple and cost-efficient 2D manufacturing process. However, the restrictions on shape and the complex mapping between parameters and shape make navigating this space a challenging task without assistance. In seeking a computational tool to help with the design process, a central question is the balance between control and automation. At one extreme of the spectrum, a fully-automated solution where the user provides a target shape and the KPS is determined through simulation and optimization requires the least amount of user intervention. However, the restricted shape of KPS will often require compromising between pure approximation quality and aesthetic considerations, which are difficult to quantify and automate. At the other end of the spectrum, manual exploration of the parameter space affords a maximum degree of artistic freedom. However, the nonlinear and unintuitive relation between parameters and shape can make manual design a tedious and frustrating process. In seeking a middle ground between those extremes, we opt for a primarily user-guided, but computer-assisted approach to shape exploration. In particular, the user is in charge of creating the structure and shape of the KPS, but can draw from a number of editing and visualization tools that simplify the design task. While this forward design approach is seemingly simple, it relies heavily on simulation and optimization to implement inverse modeling and visualization tools. The computational basis for our approach is described next.

6.3 Computational model

In order to enable computer-assisted design of KPS with desired shapes, we require a computational model for predicting (i) the equilibrium configuration for given parameter values, and (ii) the effect that parameter changes have on equilibrium shape. In this section, we describe how we define such computational model following the framework introduced in chapters 3 and 4.

6.3.1 Mechanical model

We model tensile structures as the coupling between a discrete rod mesh and thin-shell models, as they have been described in chapter 3. Next, we provide a brief review of their formulation:

- **Discrete rod mesh**. In this case, we adopt the version of the rod mesh model that explicitly considers rotational DoFs to track the orientation of the junctions, which was introduced in 3.2.2. Recall that the deformation of the rod mesh is characterized by three sets of DoFs: the positions of rod centerline $\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_{N_r})^T$, $\mathbf{c}_i \in \mathcal{R}^3$, defined at N_r rod nodes; roll angles $\theta = (\theta_1, \ldots, \theta_{N_e})^T$, $\theta_j \in \mathcal{R}$, between reference and material frames defined at N_e rod edges; and rotations of N_j junctions $\mathbf{e} = \left(\mathbf{e}_1, \ldots, \mathbf{e}_{N_j}\right)^T$ represented using Euler angles $\mathbf{e}_k \in \mathcal{R}^3$ [50]. Once again, the deformation properties of the rods are completely characterized by the set of radius $\mathbf{r} = \{\mathbf{r}_1, \ldots, \mathbf{r}_{N_r}\}$, $\mathbf{r}_i = (r_i^b, r_i^n) \in \mathcal{R}^2$ defined at rod nodes.

- **Discrete thin shells**. We model the membrane from nonlinear continuum mechanics discretized using linear triangle finite elements. In particular, we use a St. Venant-Kirchhoff (StVK) material model to compute the internal forces of the deformed fabric. We found this simple nonlinear model sufficiently accurate for our purposes since, for the kind of material and stretch factors we aim for ($\approx 50\%$), the stress-strain relation is barely nonlinear (see, e.g., Miguel et al. [148]). It should be noted that in StVK, stress is a linear function of deformation, but deformations are nonlinear w.r.t. geometry (Green strain). Although the material that we have used is almost isotropic, our implementation uses an orthotropic StVK model [64] in order to accommodate other types of fabrics, if desired. Since the pre-stretch of the fabric is significant, internal forces are strongly dominated by membrane forces, and bending forces are negligible. However, to avoid numerical problems when fabric sags, we add weak bending forces to our membrane model based on discrete shells [63]. In practical terms, the deformed configuration of the fabric is characterized by the position of the N_m membrane nodes $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_{N_m})^T$, $\mathbf{x}_i \in \mathcal{R}^3$.

We additionally have to account for coupling between the membrane and the rods embedded in it. Our approach is conceptually similar to the ones described in [14] and [16], but our specific problem and choice of discretization lead to several differences that we describe below. For the sake of computational efficiency and in order to achieve good shape approximation even for coarse meshes, we opt for an implicit coupling approach through collocation, i.e., by sharing DoFs between rod and membrane vertices. To this end, we ask that the triangle mesh representing the membrane be conforming to the embedded rods such that, for each rod vertex, there is a collocated membrane vertex. Consequently, rod mesh nodes are contained in the set of membrane nodes $N_r \subset N_m$ and the complete vector of generalized coordinates representing the deformation space is $\mathbf{q} = (\mathbf{x}, \theta, \mathbf{e}) \in \mathcal{R}^n$.

6.3.2 Design space

As it is defined in section 4.1.1, the state of the design is characterized by three sets of variables: design parameters $\mathbf{p}$, undeformed configuration $\bar{\mathbf{q}}$, and the deformed configuration $\mathbf{q}$. The design parameters uniquely define the undeformed configuration, from which the deformed configuration results through force-equilibrium constraints. Note that due to our planar fabrication process, we parameterize the rest state of both the membrane and the centerlines of embedded rods in two-dimensional space, respectively $\bar{\mathbf{x}} \in \mathcal{R}^{2N_m}$ and $\bar{\mathbf{c}} \in \mathcal{R}^{2N_r}$.

The definition of the design space is directly related with the characteristics of our fabrication method. It allows us to control four properties: i) the global stretch of the membrane, ii) the topology of the rod mesh, iii) the geometry of the rod mesh, and iv) the width and thickness of the individual rods. Formally, we parameterize each of these four properties as follows:

- We define *rod mesh topology* by a set of rods $\mathcal{R}$ and a set of junctions $\mathcal{J}$.
- For ease of fabrication, we assume uniform isotropic *membrane stretch* $s \in \mathcal{R}$.
- We define *rod geometry* by the positions $\bar{\mathbf{c}}_{c,i} \in \mathcal{R}^2$ of a set of C rod control points, which determine the positions of rod junctions as well as the shapes of individual rods.
- *Rod cross-sections* are characterized by radii values $\mathbf{r}_{c,i} \in \mathcal{R}^2$ at the same rod control points, with $\mathbf{r}_{c,i} = \{r^b_{i,c}, r^n_{i,c}\}$ corresponding to in-plane and out-of-plane directions, respectively.

Similarly to the design problem described in chapter 5, we seek for a compact and naturally smooth parameterization of the design space. With that in mind, we define the radii of the rod mesh, and the undeformed configuration of the rod mesh and the membrane by interpolating the control points defined above.

Rod mesh interpolation

For convenience in editing and optimization, we represent rods in the undeformed state as Catmull-Rom splines that interpolate the rod control points $\bar{\mathbf{c}}_c$. We evaluate each rod spline at a number of equidistant locations in spline parameter space, yielding the set of N_r rod vertices that define the piece-wise linear curves for the discrete rod model. For a given set of rod control points, the coefficients used for interpolation are precomputed; therefore the rest positions of rod vertices are linear functions of the positions of rod control points, i.e., $\bar{\mathbf{c}} = \mathbf{Z}\,\bar{\mathbf{c}}_c$, with $\mathbf{Z}$ a constant matrix of Hermite basis function evaluations. Similarly, the same interpolation is used for the definition of smooth cross-section radii at rod vertices, resulting in $\mathbf{r} = \mathbf{Z}\,\mathbf{r}_c$.

Membrane interpolation

Considering that the rod mesh and thin-shell mechanical models are coupled through collocation, during design operations, the change of rod vertex undeformed positions must effectively induce a change in the rest configuration of the membrane mesh. To ensure sufficient mesh quality for simulation, we define the rest positions for the membrane vertices implicitly as a function of the rest positions of the rod vertices. To this end, we define the vector of rest positions of the N_m membrane vertices as $\bar{\mathbf{x}} = \frac{1}{s}(\bar{\mathbf{b}}, \bar{\mathbf{c}})$, where $\bar{\mathbf{b}}$ are the $N_m - N_r$ membrane vertices that are not coincident with rod vertices, and $\bar{\mathbf{c}}$ are the N_r rod vertices. Note that the factor s is added to account for the pre-stretch of the membrane in design space. Given an initial membrane mesh and a set of rod vertices that define its boundary, we compute smoothly distributed rest positions $\bar{\mathbf{b}}$ using harmonic interpolation. The interpolation weights are computed by requiring vanishing Laplacian coordinates [190], which yields a linear system $\mathbf{L}_b\,\bar{\mathbf{b}} + \mathbf{L}_c\,\bar{\mathbf{c}} = \mathbf{0} \Rightarrow \bar{\mathbf{b}} = -\mathbf{L}_b^{-1}\,\mathbf{L}_c\,\bar{\mathbf{c}}$, with $(\mathbf{L}_b, \mathbf{L}_c)$ the Laplacian matrix. In practice, we compute cotangent weights for the Laplacian once at initialization, leaving them constant until remeshing is necessary. These constant Laplacian weights also translate into constant harmonic interpolation weights. Whenever the quality of any membrane triangle falls below a given threshold, we remesh using CGAL's Delaunay triangulation algorithm.

Final parameterization

The final parameter set $\mathbf{p} = (\mathbf{r}_c, \bar{\mathbf{c}}_c, s) \in \mathcal{R}^m$ allows us to smoothly modify the design of the KPS and determine its deformation behavior in a coherent manner, just by changing rod mesh control points and the pre-stretch coefficient. More precisely, it is worth mentioning that, thanks to our choice of discretization, membrane rest-state positions $\bar{\mathbf{x}}$, rod mesh rest-state positions $\bar{\mathbf{c}}$, and rod mesh radii $\mathbf{r}$, all linearly depend on the design parameters $\mathbf{p}$:

$$\mathbf{r} = \mathbf{Z}\,\bar{\mathbf{r}}_c \qquad \bar{\mathbf{c}} = \mathbf{Z}\,\bar{\mathbf{c}}_c \qquad \bar{\mathbf{x}} = \frac{1}{s}(\bar{\mathbf{c}}, -\mathbf{L}_b^{-1}\,\mathbf{L}_r\,\bar{\mathbf{r}}) \tag{6.1}$$

In section 6.4, we describe several tools for exploring the design space of KPS as defined by the above parameterization, based on the sensitivity analysis of the equilibrium configuration.

6.4 Computational design

Building on the computational model described in the previous section, we introduce several design tools that allow users to explore the expressive space of KPS, and to edit their shape and structure according to aesthetic considerations. In particular, we describe forward design tools for topology and base shape editing, a sensitivity analysis tool for design space exploration, and an inverse design tool for direct shape editing. These design tools make use of three major mathematical elements:

1. Fast evaluation of equilibrium KPS shapes resulting from parametric edits.

2. Sensitivity analysis for fast mapping from parametric edits to shape changes.

3. A two-step optimization algorithm that interactively solves inverse design.

These mathematical elements heavily rely on the sensitivity analysis of the equilibrium configuration that we described in detail in section 4.3.1. Roughly, the shape of a KPS is governed by equilibrium constraints $\mathbf{f}_\mathbf{q}(\mathbf{q}, \mathbf{p}) = -\nabla_\mathbf{q} V = 0$, where V is the summation of all the potential energies acting on the system. Given values for the design parameters $\mathbf{p} = (\mathbf{r}_c, \bar{\mathbf{c}}_c, s)$, we compute the corresponding equilibrium state of the KPS by solving this equation using a standard Newton's method with line search and adaptive regularization for increased robustness, as shown in section 3.4. For our parameterization, the expression corresponding to equation (4.18) for the computation of the sensitivity matrix $\mathbf{S}$ becomes

$$\mathbf{S} = \frac{\partial \mathbf{q}}{\partial \mathbf{p}} = -\frac{\partial \mathbf{f}_\mathbf{q}}{\partial \mathbf{q}}^{-1} \left(\frac{\partial \mathbf{f}_\mathbf{q}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{p}} + \frac{\partial \mathbf{f}_\mathbf{q}}{\partial \bar{\mathbf{c}}} \frac{\partial \bar{\mathbf{c}}}{\partial \mathbf{p}} + \frac{\partial \mathbf{f}_\mathbf{q}}{\partial \bar{\mathbf{x}}} \frac{\partial \bar{\mathbf{x}}}{\partial \mathbf{p}} \right). \tag{6.2}$$

In order for the above expression to be valid, the force Jacobian $\nabla_\mathbf{q} \mathbf{f}_\mathbf{q}$ has to be invertible. We eliminate null-spaces due to rigid transformations in both $\nabla_\mathbf{q} \mathbf{f}_\mathbf{q}$, $\nabla_{\bar{\mathbf{c}}} \mathbf{f}_\mathbf{q}$ and $\nabla_{\bar{\mathbf{x}}} \mathbf{f}_\mathbf{q}$ by constraining a small set of (user-selected) vertices in the deformed and rest configurations, respectively. If the system remains singular or indefinite, we add a diagonal regularizer and iteratively increase its weight until the linear solver succeeds. Due to the linear relationship defined in (6.1), it is straightforward to compute the derivatives with respect to design parameters in (6.2).

6.4.1 Forward design

During forward design, the user directly modifies the parameters listed in section 6.3, and the resulting KPS is computed automatically using the equilibrium solver discussed in section 3.4. This design approach proves most useful during the initial stage of the design, when the user defines the overall desired shape, or when topology or fabric pre-stretch need to be modified to counteract structural limitations of an intermediate design. While forward design poses virtually no restrictions on the shapes that can be achieved, it provides only limited assistance to the user.

Nevertheless, we propose several guidelines based on geometric information as well as other heuristics that have proven successful in our design experience (section 6.5).

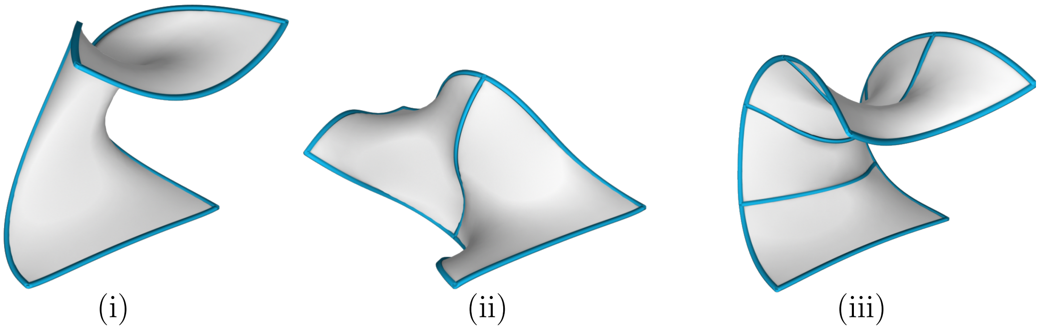


Figure 6.5: The degree of patch anisotropy affects how well a rectangular piece of fabric produces a cylindrical shape. (i) With no subdivisions, the patch curves most along the short side and the cylindrical patch appears laterally compressed. (ii) With subdivision into square patches, curvatures do not align with rods and the shape twists. (iii) With multiple subdivisions along the curved direction, the desired shape is achieved, the short rods curve as desired, and the long rods resist compression.

- **Alignment of rods with directions of principal curvature.** Since the mean curvature of the membrane vanishes everywhere, rods are the only elements whose curvature can be directly controlled to approximate the desired shape. Consequently, we generally place rods in such a way that they follow lines of principal curvatures or align with specific features.

- **Anisotropy of patches aligned with principal curvatures.** Isotropic patches with equal width in two orthogonal directions complicate the design as they favor membrane curvatures that are not aligned with rods (see Figure 6.5-ii). Anisotropic patches, on the other hand, favor curvature in the rod directions, and their shorter sides should be aligned with the direction of major curvature. Figure 6.5-iii shows a cylindrical patch subdivided along the curved direction.

- **Anisotropy of rod cross-sections.** Anisotropic rods with smaller normal width w_n than binormal width w_b favor out-of-plane over in-plane deformations; compare Figure 6.3-i and Figure 6.3-iv. Consequently, they favor the alignment of principal curvatures with rods.

- **Rod removal to relax compression.** The developability of a target design is largely determined by the rod mesh, and lack of developability can be detected by monitoring compression in the rods. When highly compressed

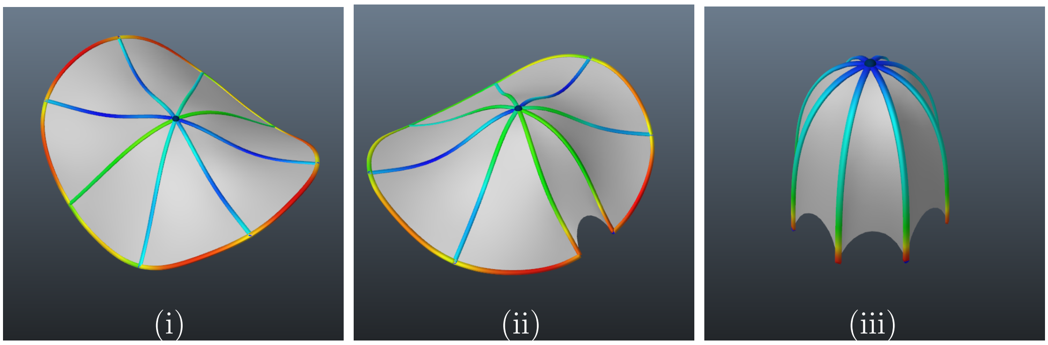


Figure 6.6: Rods are colored according to their strain with blue/red denoting highest compression/stretching. (i) Initially, radial rods are highly compressed and prevent the design from further deforming. (ii) Removal of a boundary rod relaxes compression in incident radial rods and increases the ability to bend. (iii) Removal of the complete boundary leads to non-hyperbolic shapes with high curvatures.

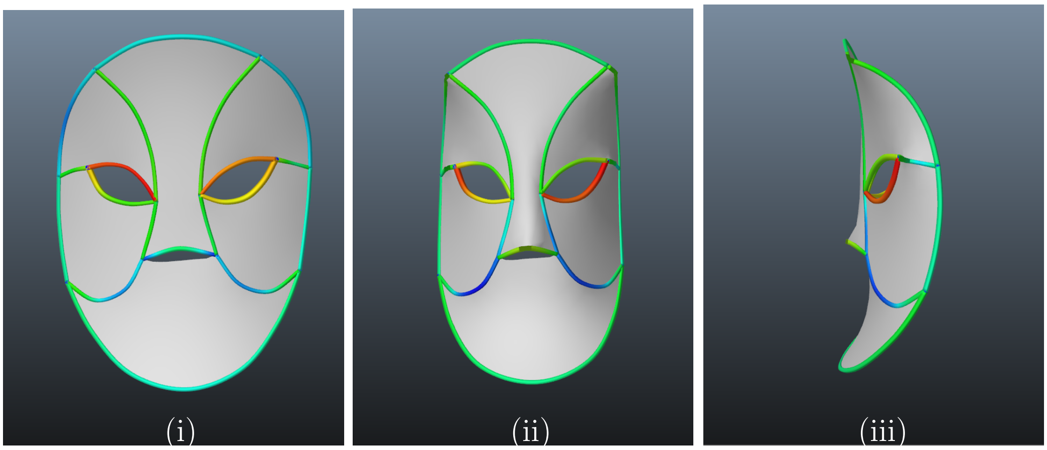


Figure 6.7: Split rods enable high-curvature. (i) Boundary rods at the top corners exhibit high compression and prevent further bending of the mask. (ii), (iii) Splitting rods at the top corners and the nose leads to overall increased curvature and enables sharp features.

rods are incident to the boundary of the design, removal of adjacent boundary rods relaxes compression and enables the exploration of more—or differently—curved shapes as shown in Figure 6.6. We implement the removal of boundary rods by transforming them into *ghost rods*, thus avoiding actual topology changes. Ghost rods are not simulated, but they are necessary for defining the control points that determine the rest positions of the membrane.

- **Rod splitting at high-curvature features.** When a rod is already bent but does not reach the desired curvature, the curvature of the membrane can be increased by splitting the rod in two. The membrane forces will produce a kink at the location of the cut, which can also be used for aesthetic purposes as shown in Figure 6.7.

- **Stretch increase to avoid sagging.** If the pre-stretch factor is not sufficiently high, the equilibrium state of the membrane can exhibit sagging and wrinkling as shown in Figure 6.3-iii. We therefore monitor and visualize the principal stretches throughout the fabric and trigger an increase in pre-stretch whenever the minimum value of any triangle element falls below a given threshold. In order to minimize deviations from the previous equilibrium shape, an increase in pre-stretch must be compensated by a corresponding increase in rod widths to withstand the larger membrane forces.

6.4.2 Sensitivity exploration

Due to the complexity of the space of KPS, design goals often fall outside the feasible space. In order to enable shape exploration while remaining within the feasible space, we visualize the design space around a given equilibrium shape. The motivating insight for this approach is that a singular value decomposition (SVD) of the sensitivity matrix reveals valuable information about the local structure of the design space. In particular, the shape changes corresponding to the *dominant*, i.e., largest, singular values convey in a concise way the major changes that a given equilibrium shape can undergo, thus offering inspiration for design changes; see Figure 6.8 for an example. In practice, analyzing the sensitivity matrix $\mathbf{S}$ alone is not sufficient, as position constraints will significantly impact the resulting modes. We therefore extend the analysis as follows.

Hard constraints

During the design process, it is often convenient to prevent parts of the model from moving or deforming. We use hard constraints to implement design goals such as fixing parts of the surface to a support structure, or for attaching disconnected parts of the surface (see Figure 6.8). Let $\mathbf{C}(\mathbf{q}) = 0$ denote the set of all such position constraints. If the current configuration already fulfills the constraints, we require

$$\mathbf{K}\Delta\mathbf{p} = \mathbf{0} , \quad \text{with} \quad \mathbf{K} = \frac{\partial \mathbf{C}}{\partial \mathbf{q}}\frac{\partial \mathbf{q}}{\partial \mathbf{p}} . \tag{6.3}$$

To enforce hard constraints to first order during editing operations, it suffices to post-multiply the sensitivity matrix $\mathbf{S}$ by a projection matrix $\mathbf{P}$, computed as in

$$\mathbf{P} = \mathbf{I} - \mathbf{K}^T \left(\mathbf{K}\mathbf{K}^T\right)^{-1} \mathbf{K} , \tag{6.4}$$

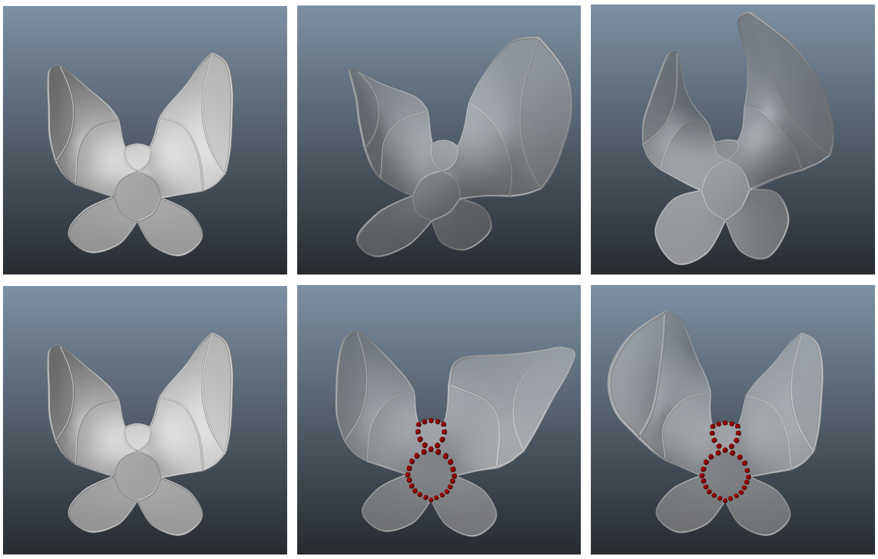


Figure 6.8: A butterfly model is deformed using the two dominant modes of the sensitivity matrix (middle and right columns). In the bottom row, hard constraints are imposed on the body of the butterfly (simulation vertices, not control points), leading to different dominant modes.

to map arbitrary parameter increments $\Delta\mathbf{p}$ to deformed state increments $\Delta\mathbf{q}$ in the null-space of the constraints as $\Delta\mathbf{q} = \mathbf{S}\,\mathbf{P}\,\Delta\mathbf{p}$. This constraint projection scheme assumes that $\mathbf{K}\mathbf{K}^T$ is invertible, which is the case if $\mathbf{S}$ is full rank and if there are no redundant constraints. It should be pointed out that this approach to enforcing hard constraints cannot remove drift, and constraint stabilization is necessary (see section 6.4.3).

Sensitivity decomposition

We incorporate the effect of hard constraints into sensitivity exploration by computing the SVD of $\mathbf{S}\,\mathbf{P}$. While we always perform the full decomposition, we generally only visualize the dominant modes corresponding to the largest singular values. For each of them, we take the corresponding right singular vector and multiply it by the sensitivity to obtain the corresponding change in shape, which is then displayed to the user (see Figure 6.8).

Using the regular sensitivity $\mathbf{S}$ in the decomposition may be misleading since the parameter vector $\mathbf{p}$ combines control point positions and cross-sectional widths, i.e., parameters with very different scales. To avoid bias, we first compute the average column norm of the sensitivity matrix for both control point and width parameters. We then normalize the columns in order to exhibit identical average norms before applying the SVD.

6.4.3 Inverse design

In addition to forward exploration of the design space, it is often convenient to directly specify desired changes to the equilibrium shape. A standard approach to implementing such *inverse design* operations is to have the user define an edit relative to a given equilibrium configuration, and to perform nonlinear optimization in order to find a feasible shape that best approximates the user-provided target.

In our setting, however, this approach is impractical: due to the restricted space of KPS, there is no guarantee that the target shape specified by the user will be close to a feasible configuration—and computing the closest feasible shape requires time-consuming nonlinear optimization. The resulting delay would interrupt the design process, and the computed shape might not meet user expectations.

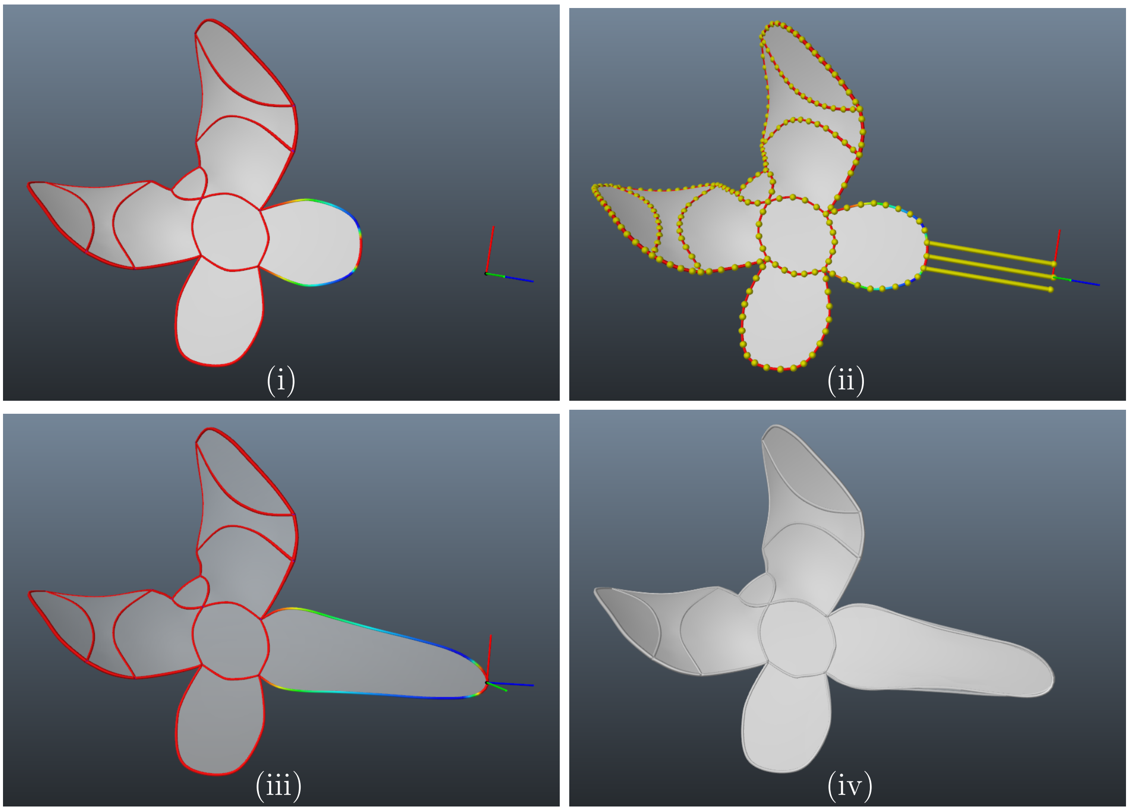


Figure 6.9: This picture shows the two-step optimization approach for user-guided inverse design. Given a base shape (i), the user defines an edit (ii) by translating a subset of the rod vertices. We interactively optimize for a target shape (iii) subject to linearized equilibrium constraints. Once the target is accepted by the user, we compute a full nonlinear optimization to obtain the final shape (iv).

Two-Step optimization

Due to the complex nature of the design problem, nonlinear optimization cannot be entirely avoided. In order to still accelerate the design process, we propose a two-step optimization approach whose goal is to increase the likelihood of the user-specified target shape to be close to feasible. In the first step, the user specifies an edited shape $\tilde{\mathbf{q}}$ starting from a given equilibrium configuration $\mathbf{q}_0$. We subsequently compute a target shape $\mathbf{q}^*$ by minimizing the distance to the user-provided edit, subject to linearized equilibrium and position constraints. While the resulting target shape will generally not satisfy the user input exactly, this optimization step is very fast, allowing for interactive and progressive exploration. Once a satisfying target $\mathbf{q}^*$ shape is found, the second step performs a fully nonlinear optimization to compute the equilibrium shape $\mathbf{q}$ that best matches the target. In this way, the time-consuming nonlinear optimization is executed only for target designs that are likely feasible.

Target optimization

In order to specify desired changes to the equilibrium shape, the user can select sets of vertices in the deformed configuration and apply rotations, translations, or scaling transformations to them (Figure 6.9-ii). We also provide a soft selection tool, allowing the user to specify per-vertex weights $w \in [0, 1]$ that determine to what extent a given vertex is influenced by the edited shape during optimization. The edited shape $\tilde{\mathbf{q}}$ defined in this way is generally infeasible, i.e., there is no choice of admissible parameter values such that the corresponding equilibrium configuration coincides exactly with the user-specified target. However, performing a nonlinear optimization to compute the closest feasible target shape would be too time-consuming for interactive exploration. We therefore use sensitivity analysis in order to compute an approximately feasible target shape that satisfies force and position constraints to first order.

Given a desired change in equilibrium shape $\Delta\tilde{\mathbf{q}} = \tilde{\mathbf{q}} - \mathbf{q}_0$, we seek to compute a change in parameters $\Delta\mathbf{p} = \mathbf{p} - \mathbf{p}_0$ that minimizes the distance measure $||\mathbf{W}\left(\Delta\tilde{\mathbf{q}} - \mathbf{SP}\Delta\mathbf{p}\right)||^2$, where $\mathbf{W}$ is a diagonal matrix of soft selection weights whose entries indicate whether –or to what extent– the position of a given vertex is defined by the editing transformation. Directly minimizing this distance measure with respect to the unknown $\Delta\mathbf{p}$ is problematic since the projected sensitivity matrix $\mathbf{SP}$ is rank-deficient. We therefore add a regularizer $||\mathbf{K}\Delta\mathbf{p}||^2$ that only penalizes parameter updates in directions that affect the position constraints.

It should be noted that, while the change in shape $\mathbf{SP}\Delta\mathbf{p}$ satisfies both equilibrium and position constraints to first order, we also have to satisfy fabrication-related constraints on the design parameters, e.g., maximum and minimum rod radii. One option to enforce the corresponding bound constraints is by constrained quadratic programming, but we found that even commercial solvers were too slow for the interactive rates required by our application. Instead of enforcing them explicitly, we therefore choose to eliminate bound constraints through parameter transformation. In essence, we introduce mapped parameters $\hat{\mathbf{p}} = \phi(\mathbf{p})$ that are asymptotically clamped to their bounds via a trigonometric transfer function; see [165] for details. Consequently, the sensitivity matrix is transformed to the new parameter space as $\hat{\mathbf{S}} = \mathbf{S}\,\nabla_p\phi$, which also affects the dependent matrices $\hat{\mathbf{K}}$ and $\hat{\mathbf{P}}$ in (6.4). Combining the above components, the target optimization problem is finally formulated as

$$\min_{\Delta\hat{\mathbf{p}}} \frac{1}{2}\left(\|\mathbf{W}\left(\Delta\tilde{\mathbf{q}} - \hat{\mathbf{S}}\,\hat{\mathbf{P}}\,\Delta\hat{\mathbf{p}}\right)\|^2 + \|\hat{\mathbf{K}}\Delta\hat{\mathbf{p}}\|^2\right), \tag{6.5}$$

where $\Delta\hat{\mathbf{p}} = (\hat{\mathbf{p}} - \hat{\mathbf{p}}_0)$. By minimizing (6.5), we obtain the optimal parameter change $\Delta\hat{\mathbf{p}}^*$ for a user-specified edit, subject to linearized equilibrium and position constraints, and we compute the first-order feasible target shape as $\mathbf{x}^* = \mathbf{x}_0 + \hat{\mathbf{S}}\,\hat{\mathbf{P}}\Delta\hat{\mathbf{p}}^*$. Moreover, since $\hat{\mathbf{S}}$ and $\hat{\mathbf{J}}$ are kept constant, (6.5) is a quadratic minimization problem that can be solved fast enough for the user to interactively explore first-order feasible edits: the user can modify the equilibrium shape until the target shape is satisfactory or the norm of the nonlinear forces exceeds a given threshold, indicating that the linear approximation is becoming invalid. In either case, the full nonlinear problem is solved with the current parameter values, and the matrices $\hat{\mathbf{S}}$, $\hat{\mathbf{K}}$, and $\hat{\mathbf{P}}$ are recomputed.

The problem (6.5) uses a linear approximation to the equilibrium manifold as described in section 4.3.3. We have seen that this approach might be prone to overshooting the step computation if the specified edit is too far away. However, it should be noted that user edits only affect a small subset of the deformation variables with high weights, and are defined incrementally from previous steps. The former implies that target optimization is most likely underdetermined, while the latter means the second-order Hessian term in (4.32) can be neglected. Consequently, in most of the cases, this linear approximation will suffice to produce good results.

Result optimization

When the first-order feasible target $\mathbf{q}^*$ is satisfactory or the residual forces are excessive, we solve the full nonlinear constrained optimization problem. The objective function is the quadratic distance from the equilibrium configuration to the target. To eliminate drift due to constraint projection, we augment the objective function with a quadratic term that penalizes the violation of the hard constraints $\mathbf{C}(\mathbf{q})$. We set a larger weight k for the hard constraints than for the target positions, but in practice, the two terms hardly interfere since the target is already first-order feasible. In addition to these objective terms, we add nonlinear force equilibrium constraints as well as bound constraints, leading to the following optimization problem:

$$\begin{aligned} \mathbf{p} &= \arg\min \frac{1}{2} \parallel (\mathbf{q}^{*} - \mathbf{q}) \parallel^{2} + \frac{1}{2}\, k\, \|\mathbf{C}(\mathbf{q})\|^{2} \\ \text{s.t.} \;\; & \mathbf{f}_{\mathbf{q}}(\mathbf{q}, \mathbf{p}) = 0 \quad \text{and} \quad \mathbf{p}_{m} \leq \mathbf{p} \leq \mathbf{p}_{M} \ . \end{aligned} \tag{6.6}$$

We solve this nonlinear optimization problem using a variant of the sensitivity-based SQP algorithm defined in section 4.3.2. To this end, we initialize the problem with the equilibrium shape $\mathbf{q}_0$ and corresponding parameter vector $\mathbf{p}_0$, then alternate between optimization steps and static equilibrium solves. In each SQP iteration, we solve the corresponding box-constrained QP using the QuickQP algorithm from the ALGLIB library. In order to efficiently compute the objective Hessian, we combine an exact Gauss-Newton-type expression for the first-order part with a BFGS-like approximation for the second-order part as described in section 4.3.3. Given a parameter update returned by the QP solver, we compute the corresponding updated shape by solving for static equilibrium.

Figure 6.9 shows an example of target and result optimization during inverse design. Our two-step optimization approach is key for achieving an interactive user experience while, at the same time, providing the accuracy demanded by the nonlinearity of the underlying design problem.

6.5 Experimental results

We have used our method to design and fabricate a set of Kirchhoff Plateau surfaces that, taken together, provide an indication of the diversity of shapes that can be achieved.

6.5.1 Printed examples

In a typical design session, the user starts with a minimum structure and then progressively modifies the topology and geometry of the rod mesh through various editing operations. Whenever needed, the user can visualize the design space using our sensitivity exploration tool and make target edits using inverse design operations. In the latter case, the user can perform standard geometric transformations on a selection of points in deformed space.

Design and simulation complexity for all the examples are summarized in Table 6.1. All designs were computed on a desktop machine with a 3.10GHz Intel i7 3770S processor with 16GB RAM. Note that target optimization was interactive for all examples. The static equilibrium solver required under one second for all examples, whereas the final result optimization took a few seconds on average.

Model	Control	Rod	Membr.	Statics	Target	Result
	points	nodes	nodes	(ms)	(ms)	(s)
Car	81	3292	4803	746	140	7.28
Mask	72	2417	3223	602	146	8.10
Flower	52	1127	2172	418	46	4.32
Butterfly	50	2351	2505	226	42	6.22
Helmet	38	1280	3522	575	33	4.13
Shoulder	37	1373	3672	333	14	5.32
Arm	28	1266	3087	311	5	3.52
Shin	46	1540	3219	510	22	3.31
Chest	88	2147	3789	545	92	6.21

Table 6.1: Summary of design complexity (number of control points), simulation complexity (number of rod and membrane nodes), and solver performance (equilibrium solver, target optimization, and result optimization) for all examples.

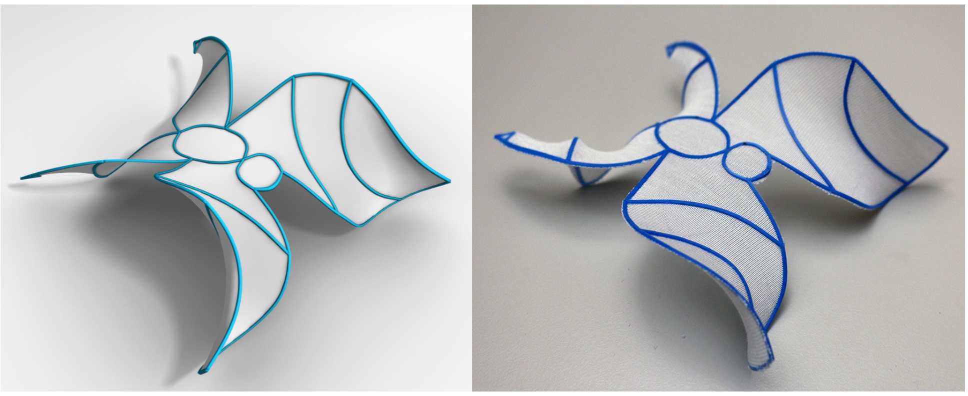


Figure 6.10: Simulated design (*left*) and physical prototype (*right*) of the butterfly example. In this example, interactive deformation feedback was key for the topology and geometry design of the overall shape. All our modeling tools were used for creating the final result.

Butterfly (Figure 6.10) The overall shape of this example was designed through forward editing of the rod topology and geometry, but the aesthetic aspects required a combination of advanced forward and inverse design tools. In particular, singular value decomposition of the sensitivity matrix revealed interesting wing deformations that were used as a basis for the final design. Inverse design operations were used for adjusting the shape and curvature of the wings.

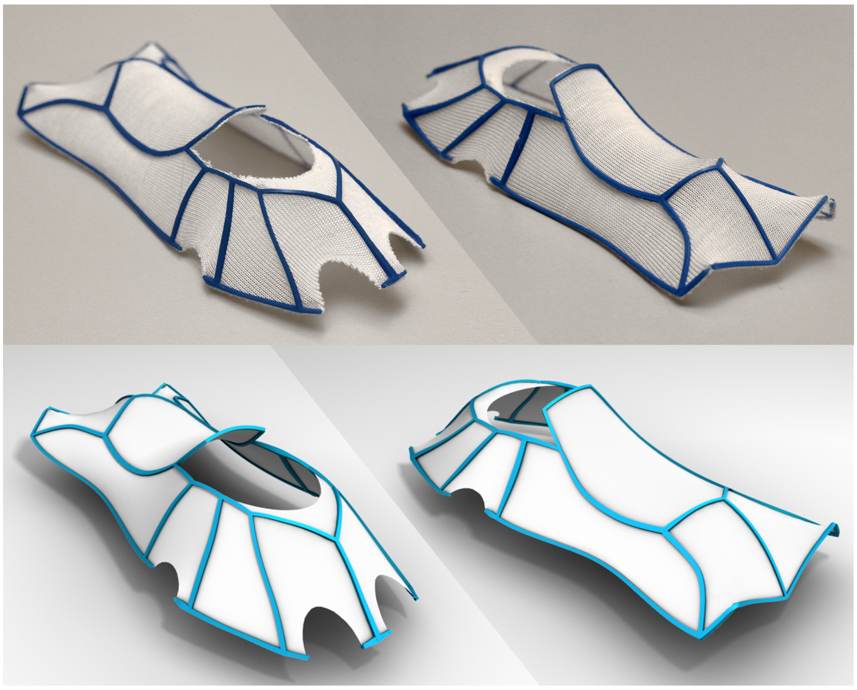


Figure 6.11: Simulated design (*bottom*) and physical prototype (*top*) of the car concept example. One of the most challenging examples due to the unintuitive and relatively dense topology and highly convex shape. Rod removal and inverse design operations were extensively used.

Car concept (Figure 6.11) As one of the most challenging examples, this design brings together complex geometric shapes, fine detail, and unintuitive topology. Most importantly, the overall shape of the target object is convex and, consequently, many of the rods were highly compressed initially. Some of the boundary rods were removed to release compression and to allow the planar rod mesh to deploy into an approximately convex shape. Inverse design operations were used to flesh out important features such as the windshield, and to adjust the size and volume of the car once the overall shape was defined. Rod splitting was also used to add detail to the front of the car, indicating the headlights.

Mask (Figure 6.12) This example combines complex internal topology with the need for large curvature in order to approximate the shape of a face. These two

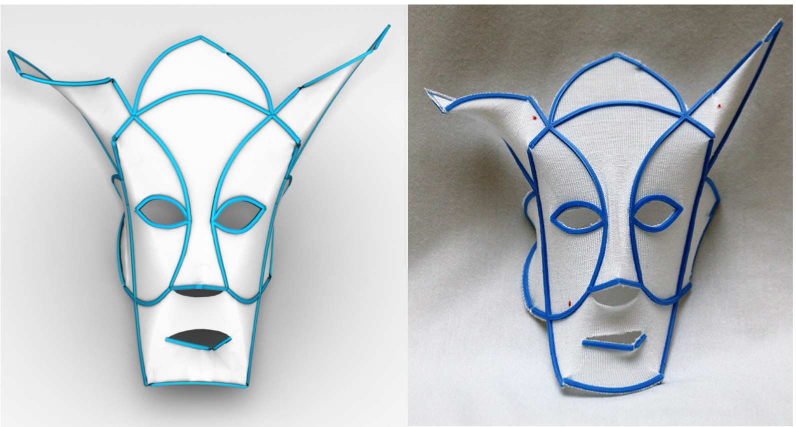


Figure 6.12: Simulated design (*left*) and physical prototype (*right*) of the mask example. In this example, facial features lead to high structural stiffness, preventing the surface from reaching the desired curvature; see also Figure 6.7-i. Rod splitting and inverse design proved essential to achieve both overall shape and details.

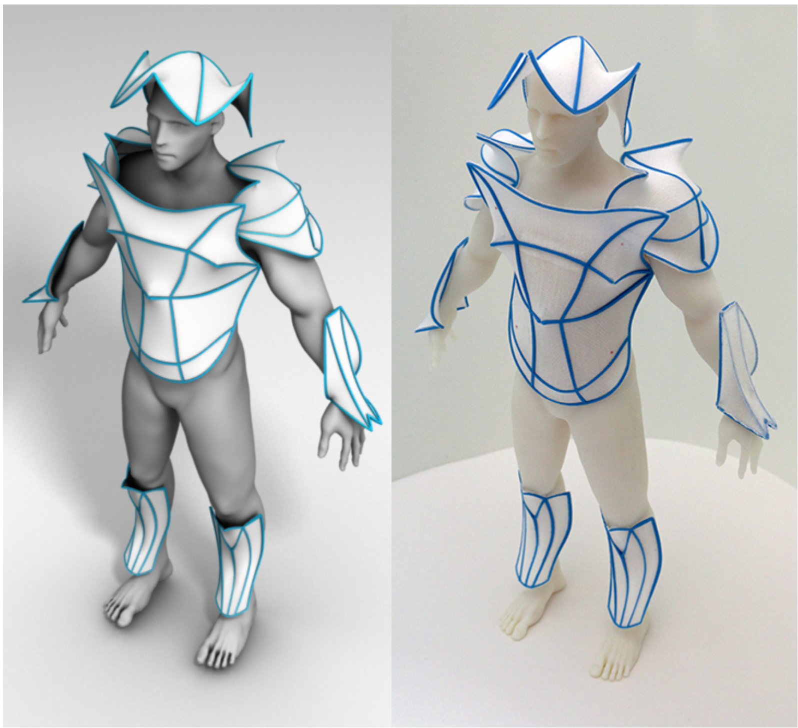


Figure 6.13: Simulated designs (*left*) and physical prototypes (*right*) of the warrior example. For this example, sensitivity exploration and inverse design were used heavily in order to adjust the size and curvature of the individual parts such as to conform to the scale of the body.

targets compete, and the complexity in topology leads to high structural stiffness which, in turn, prevents the surfaces from achieving the desired curvature. Splitting of rods at multiple points allowed the mask to curve as desired and to emphasize salient facial features such as nose and mouth. Both the overall shape and the details were edited using inverse design.

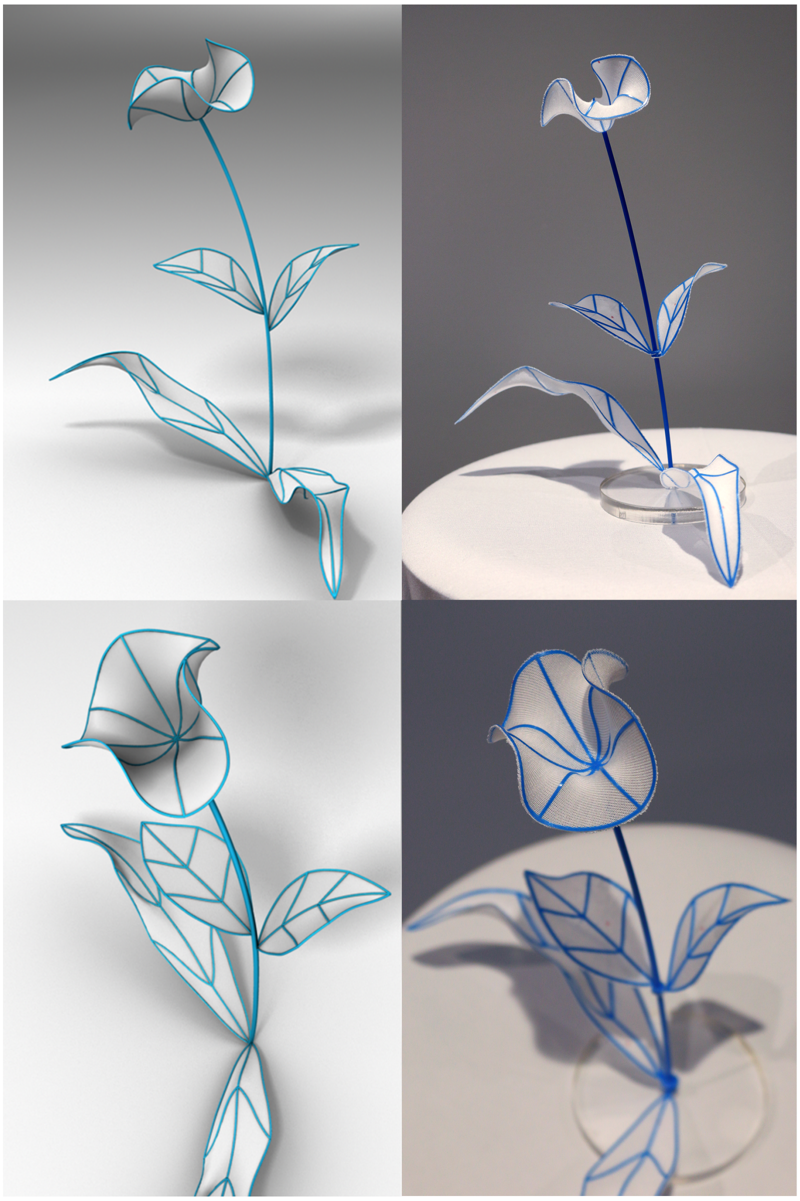


Figure 6.14: **Flower**. This design is one example of the very good agreement that we observe between the equilibrium shapes predicted by simulation (*left*) and the actual fabricated prototypes (*right*).

Warrior's armor (Figure 6.13) This composite example showcases several small parts that are geometrically and topologically simple, but the high curvature needed within small areas led to challenging fabrication constraints. Some parts such as the arms or the helmet benefited strongly from singular value decomposition of the sensitivity matrix in order to control their curvature. In addition, inverse design was used heavily to adjust the size of the different pieces to a common scale.

Flower (Figure 6.14) Another composite example, made of three parts, that illustrates the possibility to create diverse and appealing designs through a combination of forward and inverse design operations. As can be seen from the side-by-side view in Figure 6.14, this example also indicates very good correspondence between our simulation and the physically fabricated prototypes.

6.5.2 Fabrication process

Inspired by the work of Guberan and Clopath [35], we use filament-based 3D printers in order to structure stretched fabric with embedded elastic rods. Although we are primarily interested in understanding and navigating the corresponding design space, this particular manufacturing process is also a very promising way of fast and inexpensive digital fabrication. With a similar motivation, several previous works have investigated the combination of filament-based printing with other media [191], or by printing onto existing objects [192]. The fabrication process that we pursue in this work can also benefit from intelligent software for print path generation [193] and planning [30], in order to alleviate limitations of current FDM printers.

In practical terms, we used an off-the-shelf Ultimaker 2 FDM printer. The printed material is a standard PLA filament, and we used the material parameters provided by the manufacturer, i.e, Young's modulus $E = 3.31 \times 10^3 KPa$, Poisson ratio $\nu = 0.36$, density $\rho = 1240 Kg/m^3$. The membrane fabric that we used is a highly elastic, finely knitted, elastane-cotton blend. In order to obtain the material parameters for this fabric, we performed an experimental estimation of the stretch elastic moduli in both course and wale directions. We first performed several static deformation tests on a square piece of fabric, then optimized for the material parameters of a simulated counterpart. The fabric parameters obtained in this way are Young's modulus $E = 4.72 \times 10^{-2} KPa$, Poisson ratio $\nu = 0.30$, density $\rho = 122 Kg/m^3$.

Before printing, the membrane is manually stretched over a wooden frame, then clamped and inserted into the printing tray. We draw a calibration square on top of the membrane as a guideline to ensure that the resulting stretch is as uniformly distributed and isotropic as possible. For each design, the amount of pre-stretch must be chosen such as to balance fabrication constraints (e.g., minimum radii) and aesthetic considerations (e.g., absence of wrinkles). We started with a moderately low value of 40% for all examples and let the user manually increase pre-stretch if necessary.

Considering the tray size of the printer, we are restricted to small rod radii. Below some threshold, rods do not bond well to the fabric. To avoid this problem, we set a lower bound of $0.5mm$ for the cross-section. Regarding the printer settings, we set head and bed temperatures to $210°C$ and $60°C$ respectively. Finally, we used 100% flow rate and $30mm/s$ printing speed for PLA filament with $1.8mm$ diameter.

6.5.3 Discussion and future work

We have presented a method for user-guided but computer-assisted design of KPS. To circumvent the modeling challenges of KPS, we have designed simulation and optimization methods that provide the user with powerful, interactive tools that allow for intuitive exploration of the design space.

Our work is not free of limitations. First, the materials and fabrication process set important limitations on the range of results that can be achieved. As one particular example, we are currently limited by the tray size of the printer. In order to create larger models we can, in principle, combine several KPS. This, however, raises the question of how to best decompose a desired shape into pieces that can be well-approximated with KPS. The segmentation of arbitrary input models into sets of minimal surface patches is an interesting direction for future work. Other limitations are related to the modeling accuracy, both in terms of the rod and membrane models, as well as the material properties. Higher modeling accuracy would likely produce higher accuracy in the results. Finally, our method could be complemented with automatic topology optimization in order to increase the range of edits that can be achieved with inverse design operations.

Chapter 7

Conclusions

In this thesis we have presented several contributions to the field of computational design. We have proposed a novel mechanical model for the simulation of rod meshes, studied innovative optimization strategies for the solution of the elastic inverse design problem, and created computational models for the design of two different instances of rod structures: flexible rod meshes and Kirchhoff-Plateau surfaces. In this chapter, we offer general concluding remarks on each of these parts, and open a discussion about the limitations, potential impact and possible future work.

7.1 General conclusions

In this thesis, we have studied the computational design of flexible structures, a problem that poses great challenges for the development of computer-aided design tools. First, the mathematical formulation that drives the deformation behavior of the structures is often highly nonlinear, therefore efficient and accurate simulation methods are required to assess the quality of a given design. Second, the interrelation between the elements of the structure assembly hinders the creation of computational models of the design space that naturally produce good designs. And third, both previous facts lead to a highly constrained solution space whose navigation is full of perils: nonlinearities, bistable configurations, numerical ill-conditioning, etc.

We have presented innovations that contribute to the four major functionalities offered by modern computational design tools: navigating both *design* and *solution* spaces, and seamlessly moving between them through *forward* and *inverse* design tools. We have validated our contributions with the design and fabrication of two different rod structure examples that offer great application potential: flexible rod meshes and Kirchhoff-Plateau surfaces. In general, our technical solutions aim to maximize the expressive power of each specific design space while minimizing its overall complexity. The general goal is to facilitate the control of CAD tools and bring them closer to the non-expert users that will shape the future of creative design in the upcoming years.

With this purpose, we highlight the importance of creating models of the design space that are compact, tractable, expressive and capable of naturally producing coherent designs. In this sense, we have demonstrated that geometric interpolation techniques together with an adequate discretization that favors DoF collocation are simple yet powerful approaches that greatly facilitate design problems. Similarly, we also highlight the value of design guidance, adopting high-level descriptors of the deformation properties, like the example-based approach suggested in chapter 5, or providing tools to navigate the solution space like the interactive exploration method presented in chapter 6. In the following sections, we provide additional conclusions corresponding to each of the main chapters of the thesis.

7.1.1 Flexible rod meshes

In chapter 5, we have studied the computational design and fabrication of flexible rod meshes, lightweight structures whose global deformation properties can be adjusted by locally varying the radii of the cross-section of the rods. Therefore, heterogeneous deformable objects can be fabricated in one piece and from a single base material. We have proposed a simple example-based approach to exploit the expressive power of this design space. Our computational tool takes as input several poses of a deformable surface with known boundary conditions and automatically computes a single rod mesh that approximates the behavior of all the desired shapes. Our fabricated physical prototypes show that even non-expert users can easily design objects with complex deformation features –e.g., anisotropy, heterogeneity or model merging–, that can be fabricated with single-material printing machines.

7.1.2 Kirchhoff-Plateau surfaces

In chapter 6, we have studied the computational design and fabrication of Kirchhoff-Plateau surfaces, planar rod meshes embedded in pre-stretched fabric that deploy into complex three-dimensional shapes. These structures can take a rather limited range of shapes, yet they provide ample room for shape abstraction, interpretation and creativity. We have demonstrated that it is possible to navigate such a solution space by means of a user-guided but computer assisted design tool, even though it is highly non-linear, extremely constrained and full of numerical perils. The proposed catalogue of editing operations puts special attention on design guidance, featuring modal exploration of the solution space, meaningful visualization of mechanical properties, and interactive inverse design functionalities. We have validated the usefulness of these tools by designing a diverse set of complex shapes and producing physical prototypes, using a simple yet effective fabrication method that requires only a consumer level printer.

7.1.3 Mechanical simulation

In chapter 3, we have reviewed the foundations of mechanical simulation and described in detail the mechanical models and numerical solvers that we have developed in this thesis. Our main contribution in this part has been the creation of a mechanical model for the simulation of flexible rod meshes that extends the well-known reduced-model of discrete elastic rods.

Our solution considers an elastic energy that measures rod deformation w.r.t. its rigidly rotated counterpart and implicitly keeps track of connection rotations depending on the kinematic state of incident rods. Therefore it is possible to successfully transfer point forces and rotational torques between incident rods without using any additional DoF or numerical constraints. The deformation of each incident rod in the connection is affected by its own material properties and the anisotropic edge radii. This allows geometrically complex connections with heterogeneously thick incident rods to deform realistically according to the material, even for coarse discretizations.

We have defined the mechanical models used throughout this thesis to be efficient, accurate, and to lend well to numerical optimization. Therefore, we have adopted collocation of DoFs as our main approach for coupling structural components and avoid the use of numerical constraints that might obstruct the solution of the inverse elastic design problem. Our results in chapters 5 and 6 demonstrate that, with our mechanical models, it is is possible to compute the static equilibrium of sufficiently complex structures within the time budget that is available in an interactive CAD solution, and still obtain a good agreement with the behavior of physical prototypes.

7.1.4 Inverse elastic design

In chapter 4, we have presented a formal characterization of the inverse elastic design problem, we have described different numerical solving methods, and we have analyzed the challenges of their practical implementation.

In our solution, we turn away from generic constrained optimization algorithms and base our method on the iterative exploration of the equilibrium constraint manifold using sensitivity analysis. We have shown that this approach offers computational advantages related with the scalability of the solution when multiple target poses are considered. Moreover, each step of this iterative method is a fabricable solution, and therefore it is well aligned with our valued concept of guided design. Built upon this idea, we propose optimization strategies that have proved their usefulness for the problems presented in chapters 5 and 6. In the context of flexible rod meshes, we fight local minima by partitioning the parameter space depending on the sensitivity of the objective function to the variables; then, we propose a multi-resolution optimization scheme that adaptively increments the detail of the design space using interpolation. In the context of Kirchhoff-Plateau surfaces, we deal with the extremely constrained problem using a two-step optimization algorithm. First, the user interactively explores the linearized constraint manifold in order to define an approximately feasible goal. Then, the nonlinear optimization problem is solved considering this goal as the target configuration.

The performance of the sensitivity-based SQP algorithm heavily depends on two factors: i) the formulation of the quadratic subproblem and ii) the projection of approximate solutions back to the constraint manifold. We have seen that combining quasi-Newton approaches (e.g., Gauss-Newton BFGS) with a conservative step length selection (e.g., length cap, trust-region, etc.) provides satisfactory results for the kind of inverse elastic problems considered in this thesis. Built on top of this basis, we have suggested additional improvements to the QP subproblem formulation that allow us to efficiently handle parameter bounds and linear design constraints. Results in chapter 6 show that, using these techniques, it is actually possible to interactively explore the solution space even for relatively complex structures.

7.2 Discussion and future work

In this section, we open a discussion about applications, current limitations, and potential future work, related with specific topics that have been addressed throughout the thesis.

7.2.1 Rod structures applications

Assemblies of rigid or elastic rods are extensively used in structural engineering and architectural geometry due to their exceptional stiffness-to-mass ratio and low cost in terms of transport, storage, and assembly time. The computer graphics community has been recently attracted to this topic with interesting ideas for the design of tensegrities [34], reciprocal frames [33] or wire art [32] but, to this day, we believe these structures are still very underused.

Conceptually, flexible rod structures can be considered complex passive compliant mechanisms whose deformation behavior can be easily controlled by locally varying the connectivity and cross-sectional attributes of the rods. In chapter 6, we have explored one possible actuation system in the form of pre-stretched fabric, but the room is open for other alternatives like *active materials*, e.g., *shape-memory polymers* [194] or *liquid crystal elastomers* [195], that might lead to other interesting applications. Very recently, Megaro et al. [25] presented a CAD tool for replacing rigid articulations with compliant mechanisms in arbitrary mechanical systems. This idea has also received attention from the soft-robotics community where a variety of soft actuation systems are combined with –often rod-like– passive components that ultimately determine the deformation behavior, e.g., reinforced wires [196] or quadrangular rod meshes [11].

In absence of any internal actuator, designing aesthetic structures that are adaptable to an arbitrary user could be useful for many other applications like garment and furniture design. The production of personalized orthosis, i.e., wearable devices used to modify the functional characteristics of the musculoskeletal system, would benefit in particular from flexible rod meshes. A lightweight, breathable rod mesh whose deformation behavior can be easily calibrated to constrain movements in certain directions could be of great use, e.g., for the treatment of injured athletes that require a joint or body segment to be controlled, guided or immobilized.

The foremost limitation of rod structures is the current computational fabrication technology, which does not scale well with the size of the fabricated object, in terms of production costs and accessibility to the public. Nevertheless, in the realm of the small, compliant mechanisms and soft actuation technologies have also attracted the attention of the research community for the creation of microrobots [197] with applications, e.g., in biomedicine. In any of these cases, the same principles still apply and the methods developed in this thesis could be used to facilitate the design of flexible structures.

7.2.2 Inverse elastic design problem

We have shown that the sensitivity-based SQP formulation studied in chapter 4 provides satisfactory results for the inverse elastic design problems presented in this thesis. There is, however, much room for improvement in terms of performance and robustness to the numerical perils of the equilibrium constraint manifold. The quadratic subproblem often suffers from the extreme nonlinearity and ill-conditioning of the solution space and produces candidate steps that clearly overshoot the region of valid approximation. This leads to unnecessary evaluations of the objective function and costly static equilibrium computations. Considering more conservative step length selection procedures may relieve this problem but also comes with progressively moving away from quadratic convergence rates.

A possible solution might be changing the design space so that the resulting problem is better conditioned. Variable scaling is a standard approach to partially alleviate this particular problem [182], but it only affects the numerical ill-conditioning of linear system matrices, while the problem remains essentially the same. In a different direction, Musialski et al. [165] recently suggested solving the QP subproblem in a subspace of the design space in which the optimized shape properties are decorrelated. Although they obtained great improvements w.r.t. standard methods, this approach is only applicable to cases where the definition of the objective function leads to a highly underdetermined problem. In future work, we would like to explore this possibility further. One option would be to use spectral analysis on the sensitivity matrix in order to reduce the dimensionality of the problem and solve the optimization in the subspace of most sensitive directions. To our knowledge, this has not been applied yet in the context of inverse elastic design and could benefit the convergence of the problem –at least at the initial steps when the bulk fitting error has not been eliminated.

Alternatively, higher order approximations to the equilibrium constraint manifold might be also useful to reduce the deviation of candidate solutions from the nonlinear manifold. This approach would allow longer steps and alleviate the need for step length bisections. In this thesis, we have explored some approximations to the second-order Taylor expansion of the equilibrium manifold in the form of standard Newton-Raphson and quasi-Newton QP formulations, but it may be possible to find specific approaches that perform better for this particular problem. The interesting work by Yang et al. [26] also employs second-order approximants in the context of constrained mesh exploration, using quadratically parameterized osculant surfaces. However, this solution involves a costly computation of the Hessian of quadratic coefficients and the formulation does not explicitly consider the clear partition of the total space into deformation and design variables. The relationship between this geometrical interpretation and the numerical approximations considered in this thesis is intriguing and constitutes a possible line of future work.

Finally, in chapter 4, we suggested an alternative scheme for the projection of approximated steps to the constraint manifold based on the solution of the root-finding problem $\mathbf{f}(\mathbf{r}) = \mathbf{0}$ in total space. To our knowledge, this idea has not been explored yet in the context of inverse elastic design problems. Conceptually, this might open the possibility of formulating ad-hoc projection schemes that also partially minimize the design objective, hence improving the overall convergence of the problem. Our early attempts to implement this idea suggest that the approach does provide faster convergence rates in some cases, at the risk of getting slower projection times. In that sense, another possible line of future work would be to formally characterize this problem in order to overcome the numerical perils of the root-finding algorithm.

7.2.3 Design guidance and topology

We have seen that the topology of the rod structure strongly affects the solution space and hence the range of possible deformation behaviors. In this thesis, we present design approaches that avoid employing topology optimization and circumvent this limitation by using different strategies. In chapter 5, we automatically determine the hexagonal topology of the rod mesh to reduce the structural stiffness to in-plane deformation. In chapter 6, the user is directly responsible of defining the topology of the mesh, and we offer tools to visualize mechanical information that helps in the decision-making process.

In the last few years, topology optimization has attracted attention from computational design community, e.g., [10, 128]. Most of these works are based on converting the resulting discrete optimization problem into a continuous one by means of L1 regularization. This could be also applied for the design of flexible rod meshes, as the disadvantages in terms of computation time will not have much impact on an offline optimization algorithm. This extension may require a new definition of the objective function to appropriately evaluate surface deviation under arbitrary rod sampling. A different idea would be necessary for the enrichment of the interactive KPS design tool. In the context of microstructures, Schumacher et al. [12] recently suggested using topology optimization to design families of related structures that vary smoothly. A similar solution would allow our interactive tool to automatically select or suggest topology changes that could be later optimized in real time using interpolation. However, extending this idea to rod structures is very challenging. First, we would need to define a coarse-scale regular tiling so that periodic boundary conditions could be considered in the creation of structure families. Then, it would be necessary to come up with a smart parameterization of each family for the interpolation between structures, for which using signed distance fields would not be sufficient. We consider this possibility a very interesting line of future work.

Automatic topology definition could also be used for design guidance. Other applications for which high-level structural features are essential, e.g., 2D patterns [119], furniture [118] or mechanisms [158], have already considered alternative approaches for the guided exploration of the solution space. While the user focuses on aesthetics, the tool suggests discrete changes to the topology of the design whenever there is no room for improvement. These suggestions are automatically generated following heuristics, high-level functional rules or similarity measures w.r.t. a precomputed data base. In chapter 6, we introduced a series of heuristics for the forward design of KPS. In the future, it would be interesting to explore the implementation of a design tool that suggests local topology changes based on user-defined high-level goals using these heuristics.

Bibliography

[1] Romain Prévost, Emily Whiting, Sylvain Lefebvre, and Olga Sorkine-Hornung, “Make it stand: Balancing shapes for 3d fabrication,” *ACM Trans. Graph.*, vol. 32, no. 4, pp. 81:1–81:10, 2013. (Cited on pages 3, 19, and 23.)

[2] Moritz Bacher, Emily Whiting, Bernd Bickel, and Olga Sorkine-Hornung, “Spin-it: Optimizing moment of inertia for spinnable objects,” *ACM Trans. Graph.*, vol. 33, no. 4, 2014. (Cited on pages 3, 19, and 23.)

[3] Gaurav Bharaj, David I. W. Levin, James Tompkin, Yun Fei, Hanspeter Pfister, Wojciech Matusik, and Changxi Zheng, “Computational design of metallophone contact sounds,” *ACM Trans. Graph.*, vol. 34, no. 6, pp. 223:1–223:13, Oct. 2015. (Cited on pages 3 and 23.)

[4] Bernhard Thomaszewski, Stelian Coros, Damien Gauge, Vittorio Megaro, Eitan Grinspun, and Markus Gross, “Computational design of linkage-based characters,” *ACM Trans. Graph.*, vol. 33, no. 4, 2014. (Cited on pages 3 and 23.)

[5] Vittorio Megaro, Bernhard Thomaszewski, Maurizio Nitti, Otmar Hilliges, Markus Gross, and Stelian Coros, “Interactive design of 3d-printable robotic creatures,” *ACM Trans. Graph.*, vol. 34, no. 6, pp. 216:1–216:9, Oct. 2015. (Cited on pages 3, 18, 19, and 23.)

[6] Christopher Yu, Keenan Crane, and Stelian Coros, “Computational design of telescoping structures,” *ACM Trans. Graph.*, vol. 36, no. 4, pp. 83:1–83:9, July 2017. (Cited on pages 3 and 23.)

[7] 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.*, vol. 29, no. 4, pp. 63:1–63:10, 2010. (Cited on pages 4, 6, 18, and 23.)

[8] M. Skouras, B. Thomaszewski, B. Bickel, and M. Gross, “Computational design of rubber balloons,” *Comput. Graphics Forum (Proc. Eurographics)*, vol. 31, no. 2, 2012. (Cited on pages 4, 18, 19, 22, 23, 25, and 57.)

[9] Xiang Chen, Changxi Zheng, Weiwei Xu, and Kun Zhou, "An asymptotic numerical method for inverse elastic shape design," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 95:1–95:11, July 2014. (Cited on pages 4, 24, and 54.)

[10] Mélina Skouras, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, and Markus Gross, "Computational design of actuated deformable characters," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 82:1–82:10, 2013. (Cited on pages 4, 6, 19, 23, 57, and 127.)

[11] Yizhong Zhang, Like Ma, Yang Liu, Kun Zhou, and Xin Tong, "Computational design and fabrication of soft pneumatic objects with desired deformations," *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, vol. 36, no. 6, 2017. (Cited on pages 4 and 125.)

[12] Christian Schumacher, Bernd Bickel, Jan Rys, Steve Marschner, Chiara Daraio, and Markus Gross, "Microstructures to control elasticity in 3d printing," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 136:1–136:13, July 2015. (Cited on pages 4, 19, 24, and 127.)

[13] Julian Panetta, Abtin Rahimian, and Denis Zorin, "Worst-case stress relief for microstructures," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 122:1–122:16, July 2017. (Cited on pages 4 and 24.)

[14] Nobuyuki Umetani, Danny M. Kaufman, Takeo Igarashi, and Eitan Grinspun, "Sensitive couture for interactive garment modeling and editing," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 90:1–90:12, July 2011. (Cited on pages 4, 6, 18, 19, 25, 54, 61, and 103.)

[15] Bailin Deng, Sofien Bouaziz, Mario Deuss, Alexandre Kaspar, Yuliy Schwartzburg, and Mark Pauly, "Interactive design exploration for constrained meshes," *Computer-Aided Design*, vol. 61, pp. 13 – 23, 2015. (Cited on pages 4, 19, 20, 22, 26, and 54.)

[16] Mélina Skouras, Bernhard Thomaszewski, Peter Kaufmann, Akash Garg, Bernd Bickel, Eitan Grinspun, and Markus Gross, "Designing inflatable structures," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 33, no. 4, 2014. (Cited on pages 4, 6, 18, 19, 23, 25, and 103.)

[17] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer, "Elastically deformable models," in *Computer Graphics (Proc. of SIGGRAPH 87)*, July 1987, pp. 205–214. (Cited on pages 5 and 13.)

[18] D. H. House and D. E. Breen, *Cloth Modeling and Animation*, AK Peters, 2000. (Cited on pages 5 and 15.)

[19] K.-J. Choi and H.-S. Ko, "Advanced topics on clothing simulation and animation," 2005, ACM SIGGRAPH Conference Course Notes. (Cited on pages 5 and 15.)

[20] J. Spillmann and M. Teschner, "Corde: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects," in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Aire-la-Ville, Switzerland, Switzerland, 2007, SCA '07, pp. 63–72, Eurographics Association. (Cited on pages 5, 14, and 29.)

[21] Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun, "Discrete elastic rods," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 27, no. 3, 2008. (Cited on pages 5, 17, 29, 31, 32, 35, and 82.)

[22] 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 Trans. Graph. (Proc. SIGGRAPH)*, vol. 34, no. 4, 2015. (Cited on pages 5, 17, 18, 19, 35, and 61.)

[23] Jesús Pérez, Miguel A. Otaduy, and Bernhard Thomaszewski, "Computational design and automated fabrication of kirchhoff-plateau surfaces," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 62:1–62:12, July 2017. (Cited on pages 5, 17, 18, 19, 20, 35, 54, and 61.)

[24] Eder Miguel, Mathias Lepoutre, and Bernd Bickel, "Computational design of stable planar-rod structures," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 86:1–86:11, July 2016. (Cited on pages 6, 18, and 26.)

[25] Vittorio Megaro, Jonas Zehnder, Moritz Bächer, Stelian Coros, Markus Gross, and Bernhard Thomaszewski, "A computational design tool for compliant mechanisms," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 82:1–82:12, July 2017. (Cited on pages 6, 18, 19, and 125.)

[26] Yong-Liang Yang, Yi-Jun Yang, Helmut Pottmann, and Niloy J. Mitra, "Shape space exploration of constrained meshes," *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, vol. 30, no. 6, 2011. (Cited on pages 6, 19, 20, 22, 54, 61, 78, and 126.)

[27] Philippe Block and John Ochsendorf, "Thrust network analysis: A new methodology for three-dimensional equilibrium," *Journal of the International Association for Shell and Spatial Structures*, vol. 48, no. 3, 2007. (Cited on pages 7, 20, and 26.)

[28] Caigui Jiang, Chengcheng Tang, Hans-Peter Seidel, and Peter Wonka, "Design and volume optimization of space structures," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 159:1–159:14, July 2017. (Cited on pages 7, 26, and 57.)

[29] T. Van Mele, L. De Laet, D. Veenendaal, M. Mollaert, and P. Block, "Shaping tension structures by actively bent linear elements," *International Journal of Space Structures*, vol. 28, no. 3, pp. 127–135, 2013. (Cited on pages 7, 18, 20, and 25.)

[30] 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*, vol. 28, no. 3-4, pp. 187–196, 2013. (Cited on pages 7, 18, 20, 25, and 119.)

[31] 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.*, vol. 32, no. 6, pp. 177:1–177:10, 2013. (Cited on pages 7, 18, 19, 24, 26, and 57.)

[32] Lingjie Liu, Duygu Ceylan, Cheng Lin, Wenping Wang, and Niloy J. Mitra, "Image-based reconstruction of wire art," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 63:1–63:11, July 2017. (Cited on pages 7, 18, 26, and 124.)

[33] Peng Song, Chi-Wing Fu, Prashant Goswami, Jianmin Zheng, Niloy J. Mitra, and Daniel Cohen-Or, "Reciprocal frame structures made easy," *ACM Trans. Graph.*, vol. 32, no. 4, 2013. (Cited on pages 7, 26, and 124.)

[34] Nico Pietroni, Marco Tarini, Amir Vaxman, Daniele Panozzo, and Paolo Cignoni, "Position-based tensegrity design," *ACM Trans. on Graphics - Siggraph Asia 2017*, vol. 36, no. 6, Nov 2017. (Cited on pages 7 and 124.)

[35] Christophe Guberan and Carlo Clopath, "Active shoes," http://www.selfassemblylab.net/ActiveShoes.php, 2016, MIT Self Assembly Lab. (Cited on pages 8, 98, and 119.)

[36] R. E. Rosenblum, W. E. Carlson, and Tripp. E., "Simulating the structure and dynamics of human hair: modelling, rendering and animation," *J. Vis. and Comput. Anim. 2*, 1991. (Cited on page 14.)

[37] Byoungwon Choe, Min Gyu Choi, and Hyeong-Seok Ko, "Simulating complex hair with robust collision handling," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, New York, NY, USA, 2005, SCA '05, pp. 153–160, ACM. (Cited on page 14.)

[38] Andrew Selle, Michael Lentine, and Ronald Fedkiw, "A mass spring model for hair simulation," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 64:1–64:11, Aug. 2008. (Cited on page 14.)

[39] Hayley Iben, Mark Meyer, Lena Petrovic, Olivier Soares, John Anderson, and Andrew Witkin, "Artistic simulation of curly hair," in *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, New York, NY, USA, 2013, SCA '13, pp. 63–71, ACM. (Cited on page 14.)

[40] Ellis Harold Dill, "Kirchhoff's theory of rods," *Archive for History of Exact Sciences*, vol. 44, no. 1, pp. 1–23, Mar 1992. (Cited on page 14.)

[41] Dinesh K. Pai, "Strands: Interactive simulation of thin solids using cosserat models," *Proc. of Eurographics*, 2002. (Cited on page 14.)

[42] Florence Bertails, Basile Audoly, Marie-Paule Cani, Bernard Querleux, Frédéric Leroy, and Jean-Luc Lévêque, "Super-helices for predicting the dynamics of natural hair," in *Proc. of ACM SIGGRAPH '06*, 2006. (Cited on pages 14 and 29.)

[43] Sunil Hadap, "Oriented strands: Dynamics of stiff multi-body system," in *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Aire-la-Ville, Switzerland, Switzerland, 2006, SCA '06, pp. 91–100, Eurographics Association. (Cited on pages 14 and 20.)

[44] Romain Casati and Florence Bertails-Descoubes, "Super space clothoids," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 48:1–48:12, July 2013. (Cited on page 14.)

[45] M. Bergou, M. Wardetzky, S. Robinson, B. Audoly, and E. Grinspun, "Discrete elastic rods," *Proc. of ACM SIGGRAPH*, 2008. (Cited on pages 14, 15, 17, 29, 80, and 81.)

[46] M. Bergou, B. Audoly, E. Vouga, M. Wardetzky, and E. Grinspun, "Discrete viscous threads," *Proc. of ACM SIGGRAPH*, 2010. (Cited on pages 14, 29, 32, 34, 80, and 81.)

[47] J. Spillmann and M. Teschner, "Cosserat nets," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 15, no. 2, 2009. (Cited on pages 15, 17, and 35.)

[48] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff, "Position based dynamics," *Journal of Visual Communication and Image Representation*, vol. 18, no. 2, pp. 109 – 118, 2007. (Cited on pages 15 and 18.)

[49] Nobuyuki Umetani, Ryan Schmidt, and Jos Stam, "Position-based elastic rods," in *ACM SIGGRAPH 2014 Talks*, New York, NY, USA, 2014, SIGGRAPH '14, pp. 47:1–47:1, ACM. (Cited on pages 15 and 17.)

[50] Jonas Zehnder, Stelian Coros, and Bernhard Thomaszewski, "Designing structurally-sound ornamental curve networks," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 99:1–99:10, 2016. (Cited on pages 15, 17, 18, 24, 26, 35, 40, and 103.)

[51] D. Terzopoulus and K. Waters, "Analysis and synthesis of facial image sequences using physical and anatomical models," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 14, pp. 569–579, 1993. (Cited on page 15.)

[52] X. Provot, "Deformation constraints in a mass-spring model to describe rigid cloth behavior," *Proc. of Graphics Interface*, 1995. (Cited on page 15.)

[53] David E. Breen, Donald H. House, and Michael J. Wozny, "Predicting the drape of woven cloth using interacting particles," in *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1994, SIGGRAPH '94, pp. 365–372, ACM. (Cited on pages 15 and 21.)

[54] Bernhard Eberhardt, Andreas Weber, and Wolfgang Strasser, "A fast, flexible, particle-system model for cloth draping," *IEEE Comput. Graph. Appl.*, vol. 16, no. 5, pp. 52–59, Sept. 1996. (Cited on pages 15 and 21.)

[55] J. C. Simo and D. D. Fox, "On stress resultant geometrically exact shell model. part i: Formulation and optimal parametrization," *Comput. Methods Appl. Mech. Eng.*, vol. 72, no. 3, pp. 267–304, Mar. 1989. (Cited on page 15.)

[56] X.Q. Peng and J. Cao, "A continuum mechanics-based non-orthogonal constitutive model for woven composite fabrics," *Composites Part A: Applied Science and Manufacturing*, vol. 36, no. 6, pp. 859 – 874, 2005. (Cited on page 15.)

[57] Akash Garg, Eitan Grinspun, Max Wardetzky, and Denis Zorin, "Cubic shells," in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Aire-la-Ville, Switzerland, Switzerland, 2007, SCA '07, pp. 91–98, Eurographics Association. (Cited on page 15.)

[58] Bernhard Thomaszewski, Simon Pabst, and Wolfgang Strasser, "Continuum-based strain limiting," *Computer Graphics Forum*, vol. 28, no. 2, pp. 569–576, 2009. (Cited on page 15.)

[59] Olaf Etzmuβ, Michael Keckeisen, and Wolfgang Straβer, "A fast finite element solution for cloth modelling," in *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, Washington, DC, USA, 2003, PG '03, pp. 244–, IEEE Computer Society. (Cited on page 15.)

[60] Pascal Volino, Nadia Magnenat-Thalmann, and Francois Faure, "A simple approach to nonlinear tensile stiffness for accurate cloth simulation," *ACM Trans. Graph.*, vol. 28, no. 4, pp. 105:1–105:16, Sept. 2009. (Cited on page 15.)

[61] R. Goldenthal, D. Harmon, R. Fattal, M. Bercovier, and E. Grinspun, "Efficient simulation of inextensible cloth," *Proc. of ACM SIGGRAPH*, 2007. (Cited on pages 16 and 18.)

[62] Yang Liu, Hao Pan, John Snyder, Wenping Wang, and Baining Guo, "Computing self-supporting surfaces by regular triangulation," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 92:1–92:10, July 2013. (Cited on page 16.)

[63] Eitan Grinspun, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder, "Discrete shells," in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2003, SCA '03, pp. 62–67. (Cited on pages 16, 40, 44, and 103.)

[64] Yijing Li and Jernej Barbič, "Stable orthotropic materials," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2014, SCA '14, pp. 41–46. (Cited on pages 16, 40, 42, and 103.)

[65] Rahul Narain, Armin Samii, and James F. O'Brien, "Adaptive anisotropic remeshing for cloth simulation," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 152:1–152:10, Nov. 2012. (Cited on page 16.)

[66] Rahul Narain, Tobias Pfaff, and James F. O'Brien, "Folding and crumpling adaptive sheets," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 51:1–51:8, July 2013. (Cited on page 16.)

[67] Tobias Pfaff, Rahul Narain, Juan Miguel de Joya, and James F. O'Brien, "Adaptive tearing and cracking of thin sheets," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 110:1–110:9, July 2014. (Cited on page 16.)

[68] Eder Miguel, Rasmus Tamstorf, Derek Bradley, Sara C. Schvartzman, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Steve Marschner, and Miguel A. Otaduy, "Modeling and estimation of internal friction in cloth," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 212:1–212:10, Nov. 2013. (Cited on pages 16, 21, 54, and 61.)

[69] Fabian Hahn, Bernhard Thomaszewski, Stelian Coros, Robert W. Sumner, Forrester Cole, Mark Meyer, Tony DeRose, and Markus Gross, "Subspace clothing simulation using adaptive bases," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 105:1–105:9, July 2014. (Cited on page 16.)

[70] Damien Rohmer, Tiberiu Popa, Marie-Paule Cani, Stefanie Hahmann, and Alla Sheffer, "Animation wrinkling: Augmenting coarse cloth simulations with realistic-looking wrinkles," *ACM Trans. Graph.*, vol. 29, no. 6, pp. 157:1–157:8, Dec. 2010. (Cited on page 16.)

[71] Javier S. Zurdo, Juan P. Brito, and Miguel A. Otaduy, "Animating wrinkles by example on non-skinned cloth," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 1, pp. 149–158, Jan. 2013. (Cited on page 16.)

[72] Jonathan M. Kaldor, Doug L. James, and Steve Marschner, "Simulating knitted cloth at the yarn level," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 65:1–65:9, Aug. 2008. (Cited on page 16.)

[73] Jonathan M. Kaldor, Doug L. James, and Steve Marschner, "Efficient yarn-based cloth with adaptive contact linearization," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 105:1–105:10, July 2010. (Cited on page 16.)

[74] Gabriel Cirio, Jorge Lopez-Moreno, David Miraut, and Miguel A. Otaduy, "Yarn-level simulation of woven cloth," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 207:1–207:11, Nov. 2014. (Cited on pages 16 and 17.)

[75] G. Cirio, J. Lopez-Moreno, and M. A. Otaduy, "Yarn-level cloth simulation with sliding persistent contacts," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 2, pp. 1152–1162, Feb 2017. (Cited on pages 16 and 17.)

[76] Miguel A. Otaduy, Rasmus Tamstorf, Denis Steinemann, and Markus Gross, "Implicit contact handling for deformable objects," *Computer Graphics Forum*, vol. 28, no. 2, pp. 559–568, Apr. 2009. (Cited on page 17.)

[77] Jorge Gascón, Javier S. Zurdo, and Miguel A. Otaduy, "Constraint-based simulation of adhesive contact," in *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Aire-la-Ville, Switzerland, Switzerland, 2010, SCA '10, pp. 39–44, Eurographics Association. (Cited on page 17.)

[78] J. Barbic and Y. Zhao, "Real-time large-deformation substructuring," *Proc. of ACM SIGGRAPH*, 2011. (Cited on pages 17 and 38.)

[79] Duo Li, Shinjiro Sueda, Debanga R. Neog, and Dinesh K. Pai, "Thin skin elastodynamics," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 49:1–49:10, July 2013. (Cited on page 17.)

[80] Olivier Rémillard and Paul G. Kry, "Embedded thin shells for wrinkle simulation," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 50:1–50:8, July 2013. (Cited on page 17.)

[81] Zhili Chen, Renguo Feng, and Huamin Wang, "Modeling friction and air effects between cloth and deformable bodies," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 88:1–88:8, July 2013. (Cited on page 17.)

[82] Ye Fan, Joshua Litven, David I. W. Levin, and Dinesh K. Pai, "Eulerian-on-lagrangian simulation," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 22:1–22:9, July 2013. (Cited on page 17.)

[83] Shinjiro Sueda, Garrett L. Jones, David I. W. Levin, and Dinesh K. Pai, "Large-scale dynamic simulation of highly constrained strands," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 39:1–39:10, July 2011. (Cited on page 17.)

[84] Kyle Piddington, David I. W. Levin, Dinesh K. Pai, and Shinjiro Sueda, "Eulerian-on-lagrangian cloth," in *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, New York, NY, USA, 2015, SCA '15, pp. 196–196, ACM. (Cited on page 17.)

[85] Gabriel Cirio, Jorge Lopez-Moreno, and Miguel A. Otaduy, "Efficient simulation of knitted cloth using persistent contacts," in *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, New York, NY, USA, 2015, SCA '15, pp. 55–61, ACM. (Cited on page 17.)

[86] D. Baraff and A. P. Witkin, "Large steps in cloth simulation," *Proc. of ACM SIGGRAPH*, 1998. (Cited on page 17.)

[87] Pascal Volino and Nadia Magnenat-Thalmann, "Comparing efficiency of integration methods for cloth simulation," in *Proceedings of the International Conference on Computer Graphics*, Washington, DC, USA, 2001, CGI '01, pp. 265–, IEEE Computer Society. (Cited on page 18.)

[88] Sebastian Martin, Bernhard Thomaszewski, Eitan Grinspun, and Markus Gross, "Example-based elastic materials," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 72:1–72:8, July 2011. (Cited on pages 18, 19, and 20.)

[89] Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly, "Projective dynamics: Fusing constraint projections for fast simulation," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 154:1–154:11, July 2014. (Cited on page 18.)

[90] Tiantian Liu, Adam W. Bargteil, James F. O'Brien, and Ladislav Kavan, "Fast simulation of mass-spring systems," *ACM Transactions on Graphics*, vol. 32, no. 6, pp. 209:1–7, Nov. 2013, Proceedings of ACM SIGGRAPH Asia 2013, Hong Kong. (Cited on page 18.)

[91] Miles Macklin, Matthias Müller, and Nuttapong Chentanez, "Xpbd: Position-based simulation of compliant constrained dynamics," in *Proceedings of the 9th International Conference on Motion in Games*, New York, NY, USA, 2016, MIG '16, pp. 49–54, ACM. (Cited on page 18.)

[92] Huamin Wang, "A chebyshev semi-iterative approach for accelerating projective and position-based dynamics," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 246:1–246:9, Oct. 2015. (Cited on page 18.)

[93] Tiantian Liu, Sofien Bouaziz, and Ladislav Kavan, "Quasi-newton methods for real-time simulation of hyperelastic materials," *ACM Trans. Graph.*, vol. 36, no. 4, July 2017. (Cited on page 18.)

[94] Ruslan Guseinov, Eder Miguel, and Bernd Bickel, "Curveups: Shaping objects from flat plates with tension-actuated curvature," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 64:1–64:12, July 2017. (Cited on pages 18 and 25.)

[95] Desai Chen, David I. W. Levin, Wojciech Matusik, and Danny M. Kaufman, "Dynamics-aware numerical coarsening for fabrication design," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 84:1–84:15, July 2017. (Cited on page 18.)

[96] Moritz Bächer, Bernd Bickel, Doug L. James, and Hanspeter Pfister, "Fabricating articulated characters from skinned meshes," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 47:1–47:9, 2012. (Cited on pages 18 and 23.)

[97] Stelian Coros, Bernhard Thomaszewski, Gioacchino Noris, Shinjiro Sueda, Moira Forberg, Robert W. Sumner, Wojciech Matusik, and Bernd Bickel, "Computational design of mechanical characters," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 83:1–83:12, 2013. (Cited on pages 18, 19, 22, and 23.)

[98] Moritz Bächer, Stelian Coros, and Bernhard Thomaszewski, "Linkedit: Interactive linkage editing using symbolic kinematics," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 99:1–99:8, July 2015. (Cited on page 18.)

[99] Ran Zhang, Thomas Auzinger, Duygu Ceylan, Wilmot Li, and Bernd Bickel, "Functionality-aware retargeting of mechanisms to 3d shapes," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 81:1–81:13, July 2017. (Cited on pages 18 and 23.)

[100] Gerhard Dziuk and John E. Hutchinson, "The discrete plateau problem: Algorithm and numerics," *Math. Comput.*, vol. 68, no. 225, pp. 1–23, Jan. 1999. (Cited on pages 18 and 20.)

[101] Philippe Decaudin, Dan Julius, Jamie Wither, Laurence Boissieux, Alla Sheffer, and Marie-Paule Cani, "Virtual Garments: A Fully Geometric Approach for Clothing Design," *Computer Graphics Forum*, 2006. (Cited on pages 18 and 19.)

[102] Martin Kilian, Simon Flöry, Zhonggui Chen, Niloy J. Mitra, Alla Sheffer, and Helmut Pottmann, "Curved folding," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 75:1–75:9, Aug. 2008. (Cited on pages 18 and 19.)

[103] Keenan Crane, Ulrich Pinkall, and Peter Schröder, "Spin transformations of discrete surfaces," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 104:1–104:10, July 2011. (Cited on pages 18 and 20.)

[104] Hao Pan, Yi-King Choi, Yang Liu, Wenchao Hu, Qiang Du, Konrad Polthier, Caiming Zhang, and Wenping Wang, "Robust modeling of constant mean curvature surfaces," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 85:1–85:11, July 2012. (Cited on pages 18 and 20.)

[105] Chengcheng Tang, Xiang Sun, Alexandra Gomes, Johannes Wallner, and Helmut Pottmann, "Form-finding with polyhedral meshes made simple," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 70:1–70:9, July 2014. (Cited on pages 18 and 20.)

[106] Helmut Pottmann, Michael Eigensatz, Amir Vaxman, and Johannes Wallner, "Architectural geometry," *Comput. Graph.*, vol. 47, no. C, pp. 145–164, Apr. 2015. (Cited on pages 18, 20, and 26.)

[107] Marios Papas, Wojciech Jarosz, Wenzel Jakob, Szymon Rusinkiewicz, Wojciech Matusik, and Tim Weyrich, "Goal-based caustics," *Computer Graphics Forum*, vol. 30, no. 2, pp. 503–511, 2011. (Cited on page 18.)

[108] Marios Papas, Thomas Houit, Derek Nowrouzezahrai, Markus Gross, and Wojciech Jarosz, "The magic lens: Refractive steganography," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 186:1–186:10, Nov. 2012. (Cited on page 18.)

[109] Matthias B. Hullin, Ivo Ihrke, Wolfgang Heidrich, Tim Weyrich, Gerwin Damberg, and Martin Fuchs, "Computational fabrication and display of material appearance," p. 17 pages, May 2013. (Cited on page 18.)

[110] Yuliy Schwartzburg, Romain Testuz, Andrea Tagliasacchi, and Mark Pauly, "High-contrast computational caustic design," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 74:1–74:11, July 2014. (Cited on page 18.)

[111] Haisen Zhao, Lin Lu, Yuan Wei, Dani Lischinski, Andrei Sharf, Daniel Cohen-Or, and Baoquan Chen, "Printed perforated lampshades for continuous projective images," *ACM Trans. Graph.*, vol. 35, no. 5, pp. 154:1–154:11, June 2016. (Cited on page 18.)

[112] Jonàs Martínez, Jérémie Dumas, Sylvain Lefebvre, and Li-Yi Wei, "Structure and appearance optimization for controllable shape design," *ACM Trans.*

Graph. (Proc. SIGGRAPH Asia), vol. 34, no. 6, 2015. (Cited on pages 18 and 24.)

[113] Jérémie Dumas, An Lu, Sylvain Lefebvre, Jun Wu, and Christian Dick, "By-example synthesis of structurally sound patterns," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 137:1–137:12, July 2015. (Cited on pages 18 and 24.)

[114] Christian Schüller, Daniele Panozzo, Anselm Grundhöfer, Henning Zimmer, Evgeni Sorkine, and Olga Sorkine-Hornung, "Computational thermoforming," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 43:1–43:9, July 2016. (Cited on pages 18, 24, and 25.)

[115] Nobuyuki Umetani and Ryan Schmidt, "Cross-sectional structural analysis for 3d printing optimization," in *SIGGRAPH Asia 2013 Technical Briefs*, 2013. (Cited on page 18.)

[116] Marc Alexa, Kristian Hildebrand, and Sylvain Lefebvre, "Optimal discrete slicing," *ACM Trans. Graph.*, vol. 36, no. 1, pp. 12:1–12:16, Jan. 2017. (Cited on page 18.)

[117] Erva Ulu, James Mccann, and Levent Burak Kara, "Lightweight structure design under force location uncertainty," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 158:1–158:13, July 2017. (Cited on pages 18 and 25.)

[118] Nobuyuki Umetani, Takeo Igarashi, and Niloy J. Mitra, "Guided exploration of physically valid shapes for furniture design," *ACM Trans. Graph.*, vol. 31, no. 4, 2012. (Cited on pages 18, 22, and 127.)

[119] Paul Guerrero, Gilbert Bernstein, Wilmot Li, and Niloy J. Mitra, "Patex: Exploring pattern variations," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 48:1–48:13, July 2016. (Cited on pages 18, 22, and 127.)

[120] Bernd Bickel, Peter Kaufmann, Mélina Skouras, Bernhard Thomaszewski, Derek Bradley, Thabo Beeler, Phil Jackson, Steve Marschner, Wojciech Matusik, and Markus Gross, "Physical face cloning," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 118:1–118:10, 2012. (Cited on pages 19 and 24.)

[121] Jernej Barbič, Funshing Sin, and Eitan Grinspun, "Interactive editing of deformable simulations," *ACM Trans. on Graphics (SIGGRAPH 2012)*, vol. 31, no. 4, 2012. (Cited on pages 19 and 21.)

[122] S. Li, J. Huang, F. de Goes, X. Jin, H. Bao, and M. Desbrun, "Space-time editing of elastic motion through material optimization and reduction," *ACM Trans. Graph.*, vol. 33, no. 3, 2014. (Cited on pages 19 and 21.)

[123] Jiaxian Yao, Danny M. Kaufman, Yotam Gingold, and Maneesh Agrawala, “Interactive design and stability analysis of decorative joinery for furniture,” *ACM Trans. Graph.*, vol. 36, no. 2, pp. 20:1–20:16, Mar. 2017. (Cited on pages 19, 22, and 24.)

[124] 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.*, vol. 36, no. 4, pp. 157:1–157:14, July 2017. (Cited on pages 19 and 22.)

[125] Amit H. Bermano, Thomas Funkhouser, and Szymon Rusinkiewicz, “State of the art in methods and representations for fabrication-aware design,” *Computer Graphics Forum*, vol. 36, no. 2, pp. 509–535, 2017. (Cited on page 19.)

[126] Justin Solomon, Etienne Vouga, Max Wardetzky, and Eitan Grinspun, “Flexible developable surfaces,” *Comput. Graphics Forum*, vol. 31, no. 5, 2012. (Cited on page 19.)

[127] Chengcheng Tang, Pengbo Bo, Johannes Wallner, and Helmut Pottmann, “Interactive design of developable surfaces,” *ACM Trans. Graph.*, vol. 35, no. 2, pp. 12:1–12:12, Jan. 2016. (Cited on page 19.)

[128] Martin Kilian, Aron Monszpart, and Niloy J. Mitra, “String actuated curved folded surfaces,” *ACM Trans. Graph.*, vol. 36, no. 3, pp. 25:1–25:13, May 2017. (Cited on pages 19 and 127.)

[129] 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.*, vol. 35, no. 4, pp. 89:1–89:11, July 2016. (Cited on page 19.)

[130] Michael S. Floater and Kai Hormann, *Surface Parameterization: a Tutorial and Survey*, pp. 157–186, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. (Cited on page 19.)

[131] William H. III Meeks and Joaquín Pérez, “The classical theory of minimal surfaces,” *Bull. Amer. Math. Soc.*, vol. 48, pp. 325–407, 2011. (Cited on page 20.)

[132] K. Koohestani, “Nonlinear force density method for the form-finding of minimal surface membrane structures,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 6, pp. 2071 – 2087, 2014. (Cited on page 20.)

[133] L. Giomi and L. Mahadevan, “Minimal surfaces bounded by elastic lines,” *Proceedings: Mathematical, Physical and Engineering Sciences*, vol. 468, no. 2143, pp. 1851–1864, 2012. (Cited on pages 20 and 98.)

[134] Giulio G. Giusteri, Luca Lussardi, and Eliot Fried, “Solution of the kirchhoff–plateau problem,” *Journal of Nonlinear Science*, pp. 1–21, 2017. (Cited on page 20.)

[135] Yang Liu, Helmut Pottmann, Johannes Wallner, Yong-Liang Yang, and Wenping Wang, “Geometric modeling with conical meshes and developable surfaces,” *ACM Trans. Graph.*, vol. 25, no. 3, pp. 681–689, July 2006. (Cited on pages 20 and 26.)

[136] Amir Vaxman, “A projective framework for polyhedral mesh modelling,” *Comput. Graph. Forum*, vol. 33, no. 8, pp. 121–131, Dec. 2014. (Cited on pages 20 and 26.)

[137] Etienne Vouga, Mathias Höbinger, Johannes Wallner, and Helmut Pottmann, “Design of self-supporting surfaces,” *ACM Trans. Graph.*, vol. 31, no. 4, 2012. (Cited on pages 20 and 26.)

[138] Ryo Kondo, Takashi Kanai, and Ken-ichi Anjyo, “Directable animation of elastic objects,” in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA ’05)*, 2005, pp. 127–134. (Cited on page 20.)

[139] Christopher D. Twigg and Zoran Kačić-Alesić, “Optimization for sag-free simulations,” in *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, New York, NY, USA, 2011, SCA ’11, pp. 225–236, ACM. (Cited on pages 20, 24, and 54.)

[140] Alexandre Derouet-Jourdan, Florence Bertails-Descoubes, and Joëlle Thollot, “Stable inverse dynamic curves,” *ACM Trans. Graph.*, vol. 29, no. 6, pp. 137:1–137:10, Dec. 2010. (Cited on pages 20, 24, and 54.)

[141] Alexandre Derouet-Jourdan, Florence Bertails-Descoubes, Gilles Daviet, and Joëlle Thollot, “Inverse dynamic hair modeling with frictional contact,” *ACM Trans. Graph.*, vol. 32, no. 6, pp. 159:1–159:10, 2013. (Cited on page 20.)

[142] Stelian Coros, Sebastian Martin, Bernhard Thomaszewski, Christian Schumacher, Robert Sumner, and Markus Gross, “Deformable objects alive!,” *ACM Trans. Graph.*, vol. 31, no. 4, pp. 69:1–69:9, July 2012. (Cited on page 20.)

[143] Christian Schumacher, Bernhard Thomaszewski, Stelian Coros, Sebastian Martin, Robert Sumner, and Markus Gross, "Efficient simulation of example-based materials," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Aire-la-Ville, Switzerland, Switzerland, 2012, SCA '12, pp. 1–8, Eurographics Association. (Cited on page 21.)

[144] Hongyi Xu, Yijing Li, Yong Chen, and Jernej Barbič, "Interactive material design using model reduction," *ACM Trans. on Graphics*, vol. 34, no. 2, 2015. (Cited on page 21.)

[145] Hongyi Xu, Funshing Sin, Yufeng Zhu, and Jernej Barbič, "Nonlinear material design using principal stretches," *ACM Trans. on Graphics (SIGGRAPH 2015)*, vol. 34, no. 4, 2015. (Cited on page 21.)

[146] S Kawabata, *The Standardization and Analysis of Hand Evaluation (second edition)*, Hand Evaluation Standardization Committee, 11 2005. (Cited on page 21.)

[147] H. Wang, J. O'Brien, and R. Ramamoorthi, "Data-driven elastic models for cloth: Modeling and measurement," *ACM Transactions on Graphics*, 2011. (Cited on page 21.)

[148] Eder Miguel, Derek Bradley, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Miguel A. Otaduy, and Steve Marschner, "Data-driven estimation of cloth simulation models," *Comput. Graph. Forum*, vol. 31, no. 2, pp. 519–528, 2012. (Cited on pages 21, 54, 61, and 103.)

[149] Kiran S. Bhat, Christopher D. Twigg, Jessica K. Hodgins, Pradeep K. Khosla, Zoran Popović, and Steven M. Seitz, "Estimating cloth simulation parameters from video," in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Aire-la-Ville, Switzerland, Switzerland, 2003, SCA '03, pp. 37–51, Eurographics Association. (Cited on page 21.)

[150] Shoji Kunitomo, Shinsuke Nakamura, and Shigeo Morishima, "Optimization of cloth simulation parameters by considering static and dynamic features," in *ACM SIGGRAPH 2010 Posters*, New York, NY, USA, 2010, SIGGRAPH '10, pp. 15:1–15:1, ACM. (Cited on page 21.)

[151] K. L. Bouman, B. Xiao, P. Battaglia, and W. T. Freeman, "Estimating the material properties of fabric from video," in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 1984–1991. (Cited on page 21.)

[152] Hongyi Xu and Jernej Barbič, "Example-based damping design," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 53:1–53:14, July 2017. (Cited on page 21.)

[153] Mélina Skouras, Stelian Coros, Eitan Grinspun, and Bernhard Thomaszewski, "Interactive surface design with interlocking elements," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 224:1–224:7, Oct. 2015. (Cited on pages 22 and 25.)

[154] James M. Bern, Kai-Hung Chang, and Stelian Coros, "Interactive design of animated plushies," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 80:1–80:11, July 2017. (Cited on pages 22 and 25.)

[155] Sofien Bouaziz, Mario Deuss, Yuliy Schwartzburg, Thibaut Weise, and Mark Pauly, "Shape-up: Shaping discrete geometry with projections," *Comput. Graph. Forum*, vol. 31, no. 5, pp. 1657–1667, Aug. 2012. (Cited on page 22.)

[156] Niloy Mitra, Michael Wand, Hao (Richard) Zhang, Daniel Cohen-Or, Vladimir Kim, and Qi-Xing Huang, "Structure-aware shape processing," in *SIGGRAPH Asia 2013 Courses*, New York, NY, USA, 2013, SA '13, pp. 1:1–1:20, ACM. (Cited on page 22.)

[157] Youyi Zheng, Daniel Cohen-Or, and Niloy J. Mitra, "Smart variations: Functional substructures for part compatibility," *Computer Graphics Forum*, vol. 32, no. 2pt2, pp. 195–204, 2013. (Cited on page 22.)

[158] Bongjin Koo, Wilmot Li, JiaXian Yao, Maneesh Agrawala, and Niloy J. Mitra, "Creating works-like prototypes of mechanical objects," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 217:1–217:9, Nov. 2014. (Cited on pages 23 and 127.)

[159] Jacques Calì, Dan A. Calian, Cristina Amati, Rebecca Kleinberger, Anthony Steed, Jan Kautz, and Tim Weyrich, "3d-printing of non-assembly, articulated models," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 130:1–130:8, Nov. 2012. (Cited on page 23.)

[160] Duygu Ceylan, Wilmot Li, Niloy J. Mitra, Maneesh Agrawala, and Mark Pauly, "Designing and fabricating mechanical automata from mocap sequences.," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 186, 2013. (Cited on page 23.)

[161] Kiril Vidimče, Szu-Po Wang, Jonathan Ragan-Kelley, and Wojciech Matusik, "Openfab: A programmable pipeline for multi-material fabrication," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 136:1–136:12, 2013. (Cited on page 23.)

[162] Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin, "Elastic textures for additive fabrication," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 135:1–135:12, July 2015. (Cited on page 24.)

[163] Bo Zhu, Mélina Skouras, Desai Chen, and Wojciech Matusik, "Two-scale topology optimization with microstructures," *ACM Trans. Graph.*, vol. 36, no. 5, pp. 164:1–164:16, July 2017. (Cited on page 24.)

[164] Desai Chen, David I. W. Levin, Piotr Didyk, Pitchaya Sitthi-Amorn, and Wojciech Matusik, "Spec2fab: A reducer-tuner model for translating specifications to 3d prints," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 135:1–135:10, 2013. (Cited on page 24.)

[165] Przemyslaw Musialski, Christian Hafner, Florian Rist, Michael Birsak, Michael Wimmer, and Leif Kobbelt, "Non-linear shape optimization using local subspace projections," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 87:1–87:13, July 2016. (Cited on pages 24, 61, 113, and 126.)

[166] Emily Whiting, John Ochsendorf, and Frédo Durand, "Procedural modeling of structurally-sound masonry buildings," *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, vol. 28, no. 5, 2009. (Cited on page 24.)

[167] Yuki Mori and Takeo Igarashi, "Plushie: An interactive design system for plush toys," *ACM Trans. Graph. (Proc. SIGGRAPH)*, 2007. (Cited on page 25.)

[168] Aric Bartle, Alla Sheffer, Vladimir G. Kim, Danny M. Kaufman, Nicholas Vining, and Floraine Berthouzoz, "Physics-driven pattern adjustment for direct 3d garment editing," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 50:1–50:11, July 2016. (Cited on page 25.)

[169] Masaaki Miki, Takeo Igarashi, and Philippe Block, "Parametric self-supporting surfaces via direct computation of airy stress functions," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 89:1–89:12, July 2015. (Cited on page 26.)

[170] Damien Gauge, Stelian Coros, Sandro Mani, and Bernhard Thomaszewski, "Interactive design of modular tensegrity characters," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Aire-la-Ville, Switzerland, Switzerland, 2014, SCA '14, pp. 131–138, Eurographics Association. (Cited on page 26.)

[171] Akash Garg, Andrew O. Sageman-Furnas, Bailin Deng, Yonghao Yue, Eitan Grinspun, Mark Pauly, and Max Wardetzky, "Wire mesh design," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 33, no. 4, 2014. (Cited on page 26.)

[172] Henrik Zimmer and Leif Kobbelt, "Zometool rationalization of freeform surfaces," *IEEE Trans. on Visualization and Computer Graphics*, vol. 20, no. 10, 2014. (Cited on page 26.)

[173] Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun, "Discrete viscous threads," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 29, no. 4, 2010. (Cited on pages 31 and 35.)

[174] Joel Langer and David A. Singer, “Lagrangian aspects of the kirchhoff elastic rod,” *SIAM Rev.*, vol. 38, no. 4, pp. 605–618, 1996. (Cited on page 31.)

[175] Matthias Müller, B. Heidelberger, M. Teschner, and M. Gross, “Meshless deformations based on shape matching,” *Proc. of ACM SIGGRAPH*, pp. 471–478, 2005. (Cited on pages 37 and 38.)

[176] Mikhail Itskov, “A generalized orthotropic hyperelastic material model with application to incompressible shells,” *International Journal for Numerical Methods in Engineering*, vol. 50, no. 8, pp. 1777–1799, 2001. (Cited on page 41.)

[177] E. Ventsel and T. Krauthammer, *Thin Plates and Shells: Theory, Analysis, and Applications*, Marcel-Dekker, New York, New York, first edition, 2001. (Cited on page 41.)

[178] G. Irving, J. Teran, and R. Fedkiw, “Invertible finite elements for robust simulation of large deformation,” in *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Aire-la-Ville, Switzerland, Switzerland, 2004, SCA ’04, pp. 131–140, Eurographics Association. (Cited on page 44.)

[179] Eftychios Sifakis and Jernej Barbic, “Fem simulation of 3d deformable solids,” 2012, ACM SIGGRAPH Conference Course Notes. (Cited on page 44.)

[180] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H. Barr, “Discrete differential-geometry operators for triangulated 2-manifolds,” 2002. (Cited on page 44.)

[181] Ramaseshan Kannan, Stephen Hendry, Nicholas J. Higham, and Françoise Tisseur, “Detecting the causes of ill-conditioning in structural finite element models,” *Computers and Structures*, vol. 133, no. Supplement C, pp. 79 – 89, 2014. (Cited on page 47.)

[182] Arnold Neumaier, “Solving ill-conditioned and singular linear systems: A tutorial on regularization,” *SIAM Review*, vol. 40, pp. 636–666, 1998. (Cited on pages 47 and 126.)

[183] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 2nd edition, 2006. (Cited on pages 47, 48, 57, 59, 66, and 71.)

[184] David Clyde, Joseph Teran, and Rasmus Tamstorf, “Modeling and data-driven parameter estimation for woven fabrics,” in *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, New York, NY, USA, 2017, SCA ’17, pp. 17:1–17:11, ACM. (Cited on page 54.)

[185] Joseph P. Simonis, *Inexact Newton Methods Applied to Under-Determined Systems*, Ph.D. thesis, Worcester Polytechnic Institute, 2006. (Cited on page 76.)

[186] Christian Kanzow, Nobuo Yamashita, and Masao Fukushima, "Levenberg–marquardt methods with strong local convergence properties for solving nonlinear equations with convex constraints," *Journal of Computational and Applied Mathematics*, vol. 172, no. 2, pp. 375 – 397, 2004. (Cited on page 76.)

[187] Benedetta Morini and Margherita Porcelli, "Tresnei, a matlab trust-region solver for systems of nonlinear equalities and inequalities," *Comput. Optim. Appl.*, vol. 51, no. 1, pp. 27–49, Jan. 2012. (Cited on page 76.)

[188] Sébastien Valette and Jean-Marc Chassery, "Approximated centroidal voronoi diagrams for uniform polygonal mesh coarsening," *Computer Graphics Forum*, vol. 23, no. 3, pp. 381–389, 2004. (Cited on page 84.)

[189] Giulio G. Giusteri, Paolo Franceschini, and Eliot Fried, "Instability paths in the kirchhoff–plateau problem," *Journal of Nonlinear Science*, vol. 26, no. 4, pp. 1097–1132, 2016. (Cited on page 98.)

[190] Olga Sorkine, "Laplacian Mesh Processing," in *Eurographics 2005 - State of the Art Reports*, Yiorgos Chrysanthou and Marcus Magnor, Eds. 2005, The Eurographics Association. (Cited on page 105.)

[191] Stefanie Mueller, Tobias Mohr, Kerstin Guenther, Johannes Frohnhofen, and Patrick Baudisch, "fabrickation: Fast 3d printing of functional objects by integrating construction kit building blocks," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2014, CHI '14, pp. 3827–3834, ACM. (Cited on page 119.)

[192] Xiang 'Anthony' Chen, Stelian Coros, Jennifer Mankoff, and Scott E. Hudson, "Encore: 3d printed augmentation of everyday objects with printed-over, affixed and interlocked attachments," in *ACM SIGGRAPH 2015 Posters*, New York, NY, USA, 2015, SIGGRAPH '15, pp. 3:1–3:1, ACM. (Cited on page 119.)

[193] 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.*, vol. 35, no. 4, pp. 100:1–100:10, July 2016. (Cited on page 119.)

[194] Marc Behl and Andreas Lendlein, "Shape-memory polymers," *Materials Today*, vol. 10, no. 4, pp. 20 – 28, 2007. (Cited on page 125.)

[195] Ruvini Kularatne, Hyun Kim, Jennifer Boothby, and Taylor Ware, “Liquid crystal elastomer actuators: Synthesis, alignment, and applications,” vol. 55, pp. 395–411, 03 2017. (Cited on page 125.)

[196] Fionnuala Connolly, Conor Walsh, and Katia Bertoldi, “Automatic design of fiber-reinforced soft actuators for trajectory matching,” vol. 114, 12 2016. (Cited on page 125.)

[197] Lindsey Hines, Kirstin Petersen, Guo Zhan Lum, and Metin Sitti, “Soft actuators for small-scale robotics,” *Advanced Materials*, vol. 29, no. 13, pp. 1603483–n/a, 2017, 1603483. (Cited on page 125.)