INTERACTIVE DESIGN EXPLORATION OF PHYSICALLY VALID SHAPES

物理的な要求を満たす形状のインタラクティブな設計手法

by

Nobuyuki Umetani

梅谷信行

A Doctor Thesis

博士論文

Submitted to

the Graduate School of the University of Tokyo

on June 15, 2012

in Partial Fulfillment of the Requirements

for the Degree of Doctor of Information Science and Technology

in Computer Science

Thesis Supervisor: Takeo Igarashi　五十嵐健夫

Professor of Computer Science

ABSTRACT

Physical simulation makes it possible to validate geometric designs in a computer without tedious and costly physical prototyping. However, since geometric modeling and physical simulation are typically separated, simulations are mainly used for rejecting bad design, and, unfortunately, not for assisting creative exploration towards better designs. In this thesis, we propose to integrate physical simulation into geometric modeling to actively support creative design process. More specifically, we demonstrate the importance of (i) presenting the simulation results in real-time during user's interactive shape editing so that the user immediately sees the validity of current design, and to (ii) providing a guide to the user so that he or she can efficiently explore the valid deign space. To achieve these requirements, we present three algorithms each demonstrated by solid implementation of design systems with different underlying physics.

The first algorithm, "reuse of redundant intermediate data," provides the real-time response of FEM simulation with respect to the inputs of design changes. The real-time response is achieved by amortizing recompilation cost of FEM. We implemented various applications running on the system including static and dynamic solid deformation problems, fluid problems, a thermal fluid problem, and a sound wave problem. The second algorithm "first order approximation" further accelerates FEM simulation in static setting by using sensitivity analysis. It quickly predicts simulation results with respect to design changes, enabling interaction with high-resolution simulation. We demonstrate its effectiveness with a clothing design system. The third algorithm "force space analysis" is for generating useful information quickly that guides user toward physically valid design. Using the analysis in the domain of force the design system provides suggestions and annotation which tell the user how to make the model valid. We present a plank-based furniture design with nail-joint and frictional constraints as a demonstration.

These applications show the concurrent feedback and guidance from the physical simulation allow novice users to intuitively design objects with physical constraints. These algorithms have generality and can be applied to similar design support systems based on physical simulation in other domains.

論文要旨

物理シミュレーションによって，実際に試作品を作る時間と労力を掛けることなく，計算機上で手軽に設計を評価することができる．しかしながら，一般的に物理シミュレーションは形状モデリングと独立したシステム上で別々に取り扱われているために，物理シミュレーションは主に与えられた性能要求を満たすかどうかの確認にのみ使われ，創造的なデザインの過程を積極的に支援するためにはあまり使われてこなかった．本論文では，物理シミュレーションを形状モデリングに密に統合させることによって創造的なデザインを支援する手法について提案する．具体的には，この論文は，(1) ユーザが現在の設計が妥当かどうかを編集中に即座に知ることができること．(2) システムがユーザを注釈や例示によって良い設計へと誘導することを，三つのアルゴリズムを提案して実現する．各々のアルゴリズムについて，具体的な設計システムを紹介して，手法の有効性を実証する．

一つ目のアルゴリズムは中間的なデータの再利用により，設計の変更に対して，実時間で有限要素法 (FEM) のシミュレーション結果を提示するというものである．FEM 計算時に生成される中間的なデータを再利用することによって、即応性を実現している．具体例として，動的弾性体問題や，流体問題や，熱流体問題や，音場問題などへの応用例を示す．二つ目のアルゴリズムは，FEM シミュレーションの応答の一次近似を用いる物で，静的な FEM に限り，さらにシミュレーションの応答を早め，高い解像度のシミュレーションにも応用できるようにするものである．具体例として、衣服の設計システムを示す．三つ目のアルゴリズムは，ユーザーにデザインに必要となる情報を素早く提示し、物理的な制約を満たす設計へと誘導するものである．このシステムは、与えられた構造が物理的な要求を満たしているかどうかだけでなく，物理的な要求を満たすためにはその構造をどのように変形すればよいのかを，注釈と例示によって示す．例示は，設計が変わった場合にどのように構造内の力が変化するのかという感度情報を計算することにより瞬時に計算することが可能となっている．木材を釘によって接合して組み立てられる家具の設計において有用性を実証する．

これらのアプリケーションにより，即応性のあるシミュレーションからの実時間の応答や誘導によって，専門知識を持たないユーザでも簡単に物理的制約を満たす設計ができることが実証する．本論文で提案している，これらの要求機能とそれらを実現するアルゴリズムは，物理シミュレーションを用いた他のデザインシステムを設計する際にも広く応用することが可能である．

Acknowledgements

I would like to thank all those who helped me complete this work.

My utmost gratitude goes to Takeo Igarashi. No words are adequate to express my appreciation to him. He has believed in my research more than anyone else and has provided me with whatever was required to carry on with my work. He has always given me sound and accurate advice when I had problems and encouraged me to seek funding and collaboration with researchers abroad. As one of world's greatest researchers, he awakened my interest in computer graphics research and provided me with a role model. He also showed me how to work with the people in industry as a computer graphics researcher. All my research undertakings have progressed smoothly, thanks to his constant efforts. It has been my great honor to work under his supervision. I am also very grateful to my thesis committee members, Tomoyuki Nishita, Katsushi Ikeuchi, Shigeo Takahashi, Hiromasa Suzuki, and Kengo Nakajima, for providing useful suggestions for improving this thesis.

I would like to express my deep appreciation to Kinuwaki Shinichi, who has been one of my best friends since he and I were junior school classmates. He showed me some fascinating computer graphics research and motivated me to study computer graphics when I was a mechanical engineering student. He also helped me connect with other Japanese computer graphics researchers.

I would like to thank Daisuke Sakamoto for his advice on my research and on life in general and for discussions of nascent research ideas. He also provided much last-minute advice on the writing of my thesis. I spent a lot of quality time with Kenshi Takayama and am grateful for a number of insightful discussions. I also want to thank him for collaborating with me on a project involving Responsive FEM. I am thankful to many of the OB students in our laboratory, including Makoto Okabe, Ijiri Takashi, Yuki Igarashi, and Hideki Todo, for their valuable advice. I would also like to thank Jun Kato, Akira Ohgawara, Seung-Tak Noh, Yuji Yasuda, Makoto Nakajima, Shuntaro Hirakawa, Genki Furumi, Fangzhou Wang, Yuki Koyama, Naoto Sasaki, and other laboratory members for their discussions and good fellowship. I am especially grateful to Jun Kato for helping me get used to Beijing when I was at MSRA and to Seung-Tak Noh for helping me make two fast-forward videos for SIGGRAPH. I would like to thank the foreign students visiting Takeo's group, Erick Andersen and Lasse Farnung Laursen, for their companionship and for teaching me English. Especial thanks go to Erick Andersen for hosting me when I visited the University of

Washington in Seattle.

I was a member of the ERATO Design interface group, directed by Takeo Igarashi, for one and a half years. There I participated in various projects, and was inspired by a number of other great researchers. I would like to thank Mitani Jun, Yosuke Furuta, Myung Geol Choi, and Menfrad Lau for their advice and discussions.

I spent 11 months in a laboratory at Columbia University in New York City under the direction of Eitan Grinspun. I was very fortunate to have the chance to conduct research with him and the excellent post-doctoral researcher Danny Kaufman. I enjoyed my discussions with other laboratory members, including Etiena Vouga, Brenaran Smith, Brennan Smith, and Jungseock Joo. I also appreciated the companionship of Hao Li, a visitor to the laboratory. I am grateful to the Japanese researchers at Columbia University, Toshihiro Kobayashi and Hitomi Yasunobu, and to my roommates Aica Uramoto and Akiko Asanuma for our good times together in New York. My visit to Columbia University was financially supported by a Funai Oversea Scholarship. I am grateful to the Funai Foundation for Information Technology for giving me this fine opportunity, and I would especially like to thank the director, Takashi Masuda, and the founder, Tetsuro Funai.

I was very fortunate to have the chance to collaborate with the brilliant computer graphics researcher Niloy J. Mitra. He was kind enough to allow me to visit him at King Abudugh University of Science and Technology (KAUST) and University College, London (UCL), and to engage in a tight collaboration that led to a paper for SIGGRAPH 2012. I learned a lot from his enthusiastic and sincere attitude toward research and science. At UCL, I enjoyed the companionship of another visiting student, Duygu Ceylan. I also had a great time visiting the Max Plank Institute (MPI) during my visit with Prof. Niloy. I would like to thank Michael Wand of MPI for hosting me and arranging my talk and for sparing me the time for a discussion that inspired me greatly. I am very grateful to Karol Myszkowski for introducing me to a number of research projects at MPI and to Akiko Yoshida for arranging my meeting with Karol.

One of my most inspiring research experiences was working as an intern at Microsoft Research Asia (MSRA). There, I collaborated with Weiwei Xu, Xin Tong, and Jie Mao on a clothing animation project. I enjoyed my discussions and good fellowship with other members of the group, including Zhu Lifeng, Yin Yang, and Goufu Xie. I would like to thank the Japanese researchers Matsushita Yasuyuki and Koji Yatani for giving me valuable career advice. I am also grateful to my fellow interns, Hideki Ooiwa, Takeshi Sakaki, Chen Yifei, and Ying Wei, for their companionship. During my internship, I enjoyed an exciting collaboration with Teng Han and Xian Cao on their project. I was fortunate to receive a fellowship from MSRA. My internship was an option of this fellowship. I am sincerely grateful to MSRA for giving me this opportunity to participate in valuable collaborations and for financially supporting my research activity.

During my years as a Bachelor's and Master's student, I learned a lot about the

Finite Element Method (FEM)—one of main themes of this thesis—in a laboratory directed by Hisada Toshiaki, Sugiura Seiryo, Hiroshi Watanabe, and Okada Junichi. While I was working in this laboratory, Washio Takumi kindly spared me a great deal of time for discussion and taught me the mathematics of the iterative linear solver. Sawada Tomohiro helped me to understand the theory of the fluid structure interaction solver. I also benefitted from my research time with Hirabayashi Satoko, Seo Kinya, Takashi Ida, Koga Hayato, Hatano Asuka, and other students in the laboratory.

I was supported by the Mitoh Youth Program of the Information Technology Promotion Agency, Japan, when I was a Master's student. I would like to thank my project manager, Takeuchi Ikuo, for the advice he gave me on the project and on my career. It was he who encouraged me to pursue a Ph.D. degree instead of working for a company. I am grateful to the other project manager, Goto Masataka, for his advice on the metallophone design project. I would also like to thank my fellow Mitoh creators Kimihiro Mizutani, Yuji Kosuga, and Ochiai Yoichi for the good experience we shared. I am grateful to Yasutaka Sakayori for assisting and advising me on my Mitoh project.

My first research activity in a foreign country was my visit to Delft University of Technology. There, I was very well received by Kees Oosterlee and postdoctoral researcher Scott MacLachlan, who mentored me and allowed me to collaborate with them on a project involving a fast multi-grid Helmholtz solver. I would also like to thank my friends Tijmen Collignon and Medet Bazil for their kindness to me during my stay.

I am grateful to Professors Oshima Mari, Hiroshi Masuda, and Suzuki Hiromasa of the University of Tokyo for the continual career and research advice they have given me over the years.

I would like to thank my father and mother, Takehiko and Kyoko Umetani, for their steadfast support of my decisions in life.

A year and a half of my doctoral studies were funded by a fellowship from the Japanese Society for the Promotion of Science. Additionally, I received financial support from the Graduate School of Information Science and Technology at the University of Tokyo for several overseas visits.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Technological breakthroughs such as the Internet, high-performance computers, three-dimensional (3D) printing systems, and advanced software have the potential to drastically alter production and consumption practices. These technologies empower individuals to design and produce original objects, customized according to their own needs. Tailored objects can satisfy individual requirements that are economically impractical in the present system of mass production and mass consumption.

There is a growing movement (often referred to as the personal fabrication [67] or maker movement) that envisions a future in which people have their own production systems (e.g., a 3D printing system) at home or in their communities to produce customized original objects of their own design. Such a radical change in the production system could have a significant impact on society, comparable to the Industrial Revolution (which took place nearly 150 years ago).

Although this idea may sound far-fetched, the success of many emerging commercial services testifies to its future promise. Autodesk [1] and Shapeways [143] provide online services that enable end users to design objects, share them online with other users, and even place orders for them to be manufactured (Figure 1.1). In addition, various affordably priced 3D printing devices, such as Replicater, Cubify, and Solidoodle, are now on the market, targeting end users. Moreover, local facilities such as FabLab [61] offer manufacturing devices and technical support in their stores, so that end users can explore their own creativity in a casual atmosphere. FabLab is gaining popularity worldwide, and has branches in most major cities.

However, a major unresolved challenge to personal fabrication is how to support the design of visually pleasing yet functional objects by users who lack specialized engineering skills. The devices employed in personal fabrication impose few constraints on the design to be manufactured, thus offering a wide range of design possibilities to users. For example, 3D printers can print objects with complicated inner structures, and laser cutters can cut sheet material such as paper, cloth, or wood into arbitrary shapes. Fabrication devices with such capabilities endow users with endless design possibilities, and one can leverage these possibilities to fulfill both aesthetic and functional goals.

Figure 1.1: *Examples of design and creation using the Autodesk 123D [1] service.*

1.1 Existing Systems

Aesthetic goals are dominant in the design of decorative objects such as accessories, paper cutouts, and ornaments. Functional goals are important in the design of objects with physical functions such as musical instruments, clothing, and furniture. In the design of decorative objects, visual appeal is paramount. Hence, in this case, users consider only the geometric aspects of their designs, in accordance their own artistic inclinations. On the other hand, the design of functional objects requires attention to both physical properties and aesthetic appeal. For example, in the context of DIY furniture design, various physical constraints must be satisfied. (A chair is only useful if it remains stable and does not break under targeted load distributions.) In the context of clothing pattern design, the clothing (made of fabrics that are cut and stitched together) should fit the human body well. Incorporating these physical constraints into the design process is very difficult for novice users who lack specialized knowledge in the relevant fields. For this reason and others, the realization of design systems that enable novice users to design functional products remains a formidable challenge.

A number of sophisticated geometric modeling tools have been developed over the years, but none of them actively assist users in the design of physically valid objects. General-purpose geometrical modeling systems such as CATIA [37], Sketch-up [70], and Blender [26] have advanced to the point that end users can readily design objects with various shapes, based solely on their own artistic inclinations. However, it is extremely difficult for non-experts to design physically valid objects using these systems, because the systems do not provide physical information to help such users correctly understand the physical properties of the objects.

Existing geometric modeling tools are intended either for use by experts (e.g., computer-aided design [CAD] systems) or for modeling objects whose visual aspects are the only consideration (e.g., computer graphics modeling systems). Thus, a new generation of design tools is needed to support the creation of physically valid objects by novice users in the age of personal fabrication.

Computerized physical simulation makes it possible to physically validate geometric designs without tedious and costly physical prototyping. However, geometric modeling and physical simulation are typically separated, and physical plausibility is not easily incorporated into the design phase. More specifically, most numerical simulation systems (generally referred to as computer-aided engineering [CAE] tools) are used to verify designs offline, after the design process has been completed. In a typical situation, a designer creates a 3D geometric shape using a CAD program, and then validates it using a physical simulator (e.g., an FEM solver). If the shape violates one or more physical constraints, it is sent back to the designer for refinement. This process is iterated until the refinements lead to a satisfactory design.

Such a workflow causes various problems. It is time-consuming to run the simulation (Problem I), and the results provide no guidance to the designer as to how to rectify the current physical constraint violations (Problem II). Hence, the design process often involves a lot of tedious trial and error, and simulations have mainly been used for rejecting designs that fail to satisfy requirements, rather than for assisting creative exploration (see Figure **??**-Left).

Structural optimization tools are sometimes used to create physically valid designs. However, optimization tools only provide a single valid design, and thus the user cannot control the final optimized shape at will. Users lacking specialized knowledge and experience find it difficult and tedious to utilize this type of system to create interesting designs that also satisfy physical constraints.

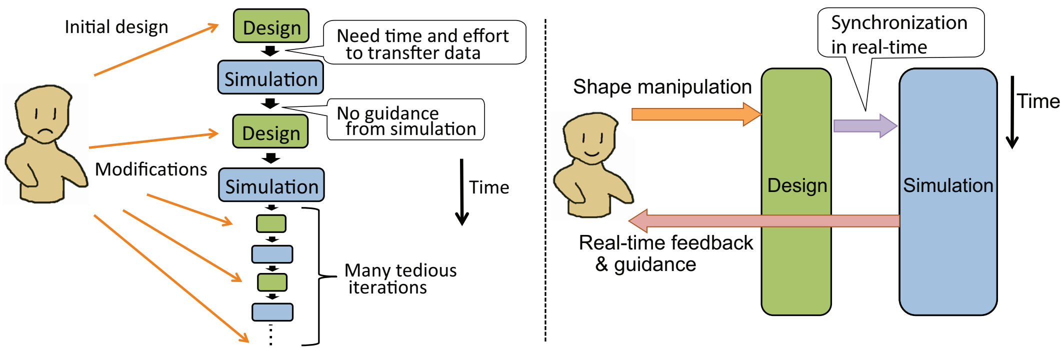


Figure 1.2: *Left: workflow of a typical separated design and simulation system which requires many trials-and-errors due to the lack of interaction. Right: workflow of our integrated design and simulation system which provides real-time simulation feedback assisting the user's creative exploration.*

1.2 Our Approaches

The present work describes our efforts to address Problems I and II by integrating physical simulation into interactive geometric modeling to actively support the creative design process. More specifically, we present systems that give real-time feedback to users, including notification of whether or not a physical constraint is satisfied. We also describe a system that helps users to maintain or restore physical validity during the interactive design process.

Computational speed has been a major obstacle to achieving real-time feedback from simulations during interactive design. Design verification often requires finite element method (FEM) analysis, which is generally too expensive for real-time computations in a changing geometry. The technical contribution of this thesis is the introduction of novel algorithms that achieve real-time FEM feedback during the design process, and provide guidance for maintaining and restoring physical validity. In the FEM framework, these algorithms reduce the computations involved in a design change. We present three systems to demonstrate the effectiveness of our algorithms, and include a separate detailed description of each of them (see Figure **??**-right).

First, we discuss a system that runs a real-time simulation in the background of the design system, so that the system continuously provides feedback on dynamic user input during geometric editing. Typically, standard real-time simulations are applied to a fixed initial geometry. In contrast, our method continuously updates the simulation results in response to user modifications of the initial geometry. It is well understood that instant simulation feedback during modeling is important for reducing the amount of trial and error. However, no previous research has presented a working algorithm that provides instant feedback. Even state-of-the-art CAD systems (e.g., CATIA) do not support continuous real-time feedback to check the validity of current designs. To tackle this problem, we propose the *reuse of redundant intermediate data* to accelerate both static and dynamic FEM simulations. FEM entails several computations, including mesh generation, matrix generation, and solving a linear system. By continuously deforming a mesh instead of rebuilding it from scratch each time, we can reuse the resultant data on features such as the matrix structure and the results of symbolic factorization. The FEM response is greatly accelerated by switching between multi-level reuse schemes according to the amount of mesh deformation.

Second, we propose a further acceleration technique based on *first-order approximation*, which is applicable to static simulations. FEM simulations are classified into static and dynamic simulations. A static simulation is used to determine the static equilibrium status, while a dynamic simulation is used to determine the transitional status. We exploit the observation that a single static simulation result is obtained as an output from an input shape, to build a continuous map between input shape and simulation result. By investigating the linear response between input shape and output static simulation result, we can approximate the simulation result produced by an input shape change. First-order approximation of a simulation response is often considered design-sensitivity analysis, which is typically used for offline shape optimization. However, we propose a way to use design-sensitivity analysis in an interactive context.

Finally, we describe a procedure that guides the development of physical designs. Specifically, the system provides annotations and suggestions, in addition to real-time simulation results. The annotations and suggestions help users efficiently explore the physically valid design space. The annotations selectively visualize useful information obtained from the real-time simulations, such as the physically valid ranges of design parameters being edited. This type of annotation relieves the user of the burden of ensuring feasibility. We also propose a methodology that provides suggestions for restoring validity when the current design is not valid. Even if there are numerous possible physically valid shapes, our algorithm provides only meaningful suggestions that are sufficiently different from each other. We propose a *force space analysis* technique, which represents the physical validity in a space of constraint forces, to quickly generate suggestions and annotations, based on the information obtained from the simulation results. Note that, in contrast to a direct optimization-based solution, we leave the designer in control of form-finding. We only provide valid range visualization and multiple deformation suggestions for a feasible geometric shape when necessary.

To demonstrate the effectiveness of these algorithms, we present various specialized design systems, targeting different physical requirements. Instant simulation feedback during interactive design via the *reuse of redundant intermediate data* techniques is effective for general FEM problems involving up to tens of thousands of degrees of freedom. We discuss several engineering design problems dealing with physical phenomena, such as thermal fluids, vibration, and wave propagation. We verify that instant feedback is helpful for designing a physically valid object, by performing a user study comparing bridge design systems with and without instant feedback. We describe a metallophone design system, and fabricate a metallophone designed using this system. The metallophone design presented here is very complicated, and is not possible without instant simulation feedback.

Then we demonstrate the *first-order approximation* acceleration technique, which can be applied to static FEM simulations involving up to a hundred thousand degrees of freedom. We demonstrate our algorithms in an interactive clothing pattern design system. Because pattern design is very popular among end users, and the resulting articles of clothing are often tested by putting them on a mannequin, it is a very suitable target application for this technique.

Guidance from the system for creating a physically valid design is studied in a nail-jointed, plank-based furniture design system. In plank-based furniture design, two widely separated planks might be structurally coupled, and hence it is very difficult for a non-expert user to design valid furniture. While larger metallophones always produce lower tones in the metallophone design system, and larger patterns always produce looser clothing in the pattern design system, the furniture design system does not present such an obvious relationship between validity and design. Hence, valid furniture design is hardly possible without guidance from the system. This system can handle rigid-body simulations with up to two hundred degrees of freedom. We demonstrate the effectiveness of our approach by comparing our system with a typical design system in which design and simulation are separated.

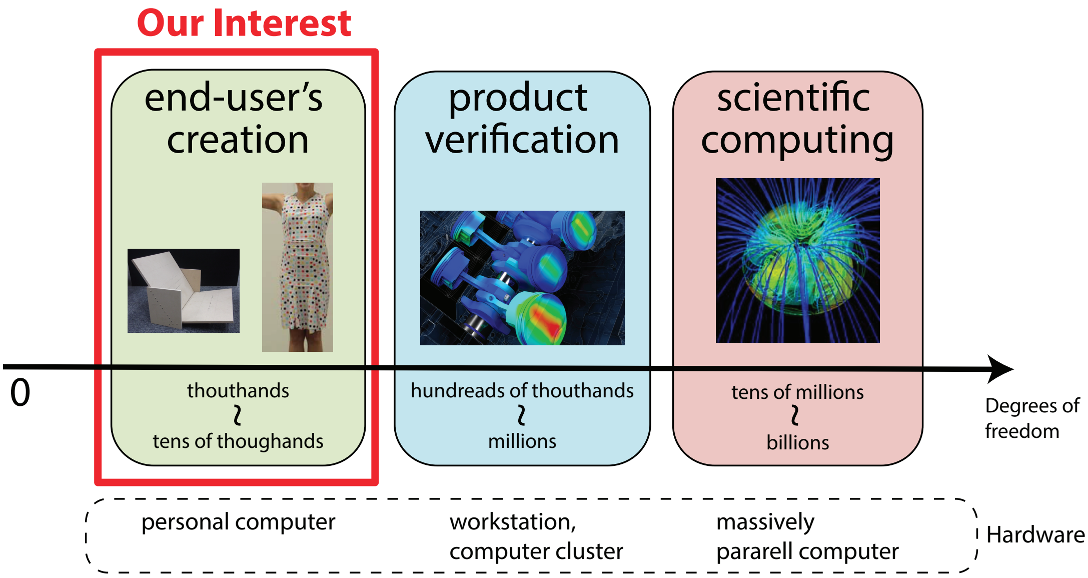


Figure 1.3: *Our scope of interest is relatively small-scale simulation enough for checking basic functionality of a designed object.*

We limit the scope of the applications in this thesis to those that involve relatively simple physics (see Figure 1.3). Accuracy and computational time have a trade-off relationship in physics simulations. Accuracy often amounts to the number of degrees of freedom used in the simulation; a highly accurate simulation requires a detailed representation of the problem. This thesis targets FEM simulations on a rather small scale, but still with enough accuracy to produce the users' creations. The accuracy required in a simulation depends on the aim of the simulation. If a simulation is connected with scientific research involving many small factors, a very large number of degrees of freedom (up to billions) may be required, and highly paralleled hardware is normally used. Such large-scale simulations take a considerable amount of time, on the order of days or even weeks. Simulations used for product verification require fewer degrees of freedom (up to hundreds of thousands). Because the targeted natural phenomena are well understood and often very well modeled, the number of degrees of freedom is smaller than that required for scientific computing. For example, if the brittle fracture of a glass pane is being studied, a scientific simulation may solve for the growth of each tiny crack, whereas an engineering simulation may employ some continuous approximations. However, even with this simplification, the simulation still takes a long time (on the order of hours). Our work targets end-user creations for which originality is more important than optimal performance. Hence, the accuracy required here is limited to checking the basic functionality of a designed object, and requires even fewer degrees of freedom (up to tens of thousands). However, such relatively small-scale simulations still take a non-negligible amount of time (several seconds or minutes), and have been impractical for interactive systems. Our algorithm brings these small-scale problems into the realm of real-time interaction.

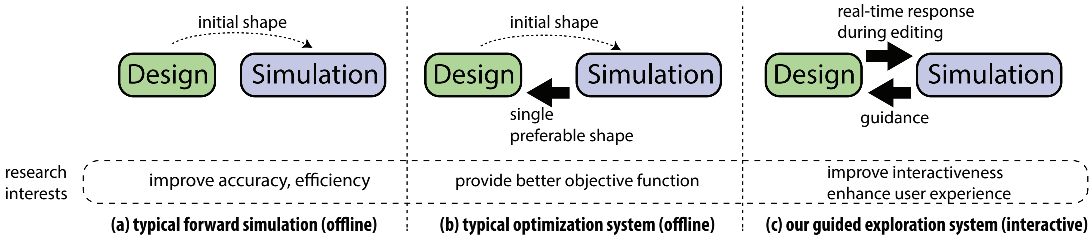


Figure 1.4: *Research interests of (a) typical forward simulations, (b) typical design optimizations, and (c) our design exploration.*

Our research interests are different from those of typical design and simulation systems. From the point of view of relationship between design and simulation, there are mainly three approaches. One is a forward simulation approach, where a offline simulation is performed with single initial design (Figure 1.4-(a)). Another is a design optimization approach, where a simulation provides a single physically valid design that is computed as similar to initial design as possible using offline computation (Figure 1.4-(b)). The other is our design exploration system, where the simulation is performed in real-time and provide guidance to the user during the user's design session (Figure 1.4-(c)). In the research of a forward simulation, the research interest is usually improving accuracy and computational efficiency while sacrificing interactiveness. The research interest of the optimization is usually finding objective function that describe user's demand well. On the other hand, our design exploration system tries to maximize the user's experience during the design session by improving interactiveness between the design and simulation.

In summary, we show that an interactive design system that supports instant simulation feedback during modeling and also provides guidance for creating a valid design is essential for exploring physically valid designs. Interaction with a physics simulation makes it possible for non-expert users to design objects reflecting their own aesthetic values, while satisfying physical constraints. Typically, obtaining simulation results while modeling is computationally expensive, and it was challenging to make the procedure interactive while maintaining an adequate problem scale for users' creations. In this thesis, we propose two acceleration algorithms: *the reuse of redundant intermediate data* and *first-order approximation*. Furthermore, a *force space analysis* algorithm makes it possible to provide guidance at an interactive speed. We believe that our interactive design and supporting algorithms have sufficient generality to be applied to many other design problems where physical validity is important.

1.3 Thesis Overview

This thesis is organized as follows:

1. In Chapter 2, we review the design workflow currently used in the industry to manufacture a product (as background material). We also discuss some applications that use real-time simulations.

2. In Chapter 3, we present an overview of recent research related to computerized design of physically valid shapes. We focus on four topics: real-time simulation, design optimization, suggestive modeling, and modeling for fabrication. For each topic, we describe the research carried out in both the computer graphics and engineering communities.

3. In Chapter 4, we discuss the integration of FEM simulations into interactive design for a wide range of applications. We describe a method of achieving responsive speed for static and dynamic FEM simulations during interactive design. We also demonstrate several systems for handling engineering problems, as well as a metallophone design system, as applications of this method.

4. In Chapter 5, we present an acceleration technique for static FEM simulations, to quickly provide simulation results in response to design changes. This technique uses design-sensitivity analysis as a first-order approximation of the simulation results. We validate the effectiveness of the algorithm in an interactive clothing pattern design system.

5. In Chapter 6, we discuss a method for providing annotations and suggestions to guide user designs. The system analyzes the current design, simulation results, and mode of user manipulation to provide useful information for the design of physically valid shapes. The system also provides suggestions for restoring physical validity. These annotations and suggestions are demonstrated in an application targeting nail-jointed, plank-based furniture. We demonstrate the effectiveness of the annotations and suggestions via an informal user study.

6. In Chapter 7, we conclude the thesis. We briefly summarize what is required to integrate simulation into a design system, as a set of design guidelines. Finally, we describe possible directions for future research and applications.

1.4 Publications

The following is a list of publications from which this thesis was derived:

1. Our system for providing real-time FEM simulation response to design change was published as "Responsive FEM for Aiding Interactive Geometric Modeling

” in the journal IEEE Computer Graphics & Applications, in collaboration with Takeo Igarashi and Kenshi Takayama from the University of Tokyo, and Jun Mitani from University of Tsukuba.

2. Our integrated system for designing a metallophone was published as “Designing Custom-made Metallophone with Concurrent Eigenanalysis,” in New Interfaces for Musical Expression++ (NIME++) 2010 in Australia, in collaboration with Takeo Igarashi from the University of Tokyo.

3. Our integrated system for clothing pattern design was published as “Sensitive Couture for Interactive Garment Editing and Modeling,” in ACM Transactions of Graphics 2011 in Vancouver, WA, USA, in collaboration with Danny M. Kaufman and Eitan Grinspun from Columbia University, and Takeo Igarashi from the University of Tokyo.

4. Our interactive system for furniture design was published in ACM Transactions of Graphics 2012 in Los Angeles, USA, as “Guided Exploration of Physically Valid Shapes for Furniture Design,” in collaboration with Takeo Igarashi from the University of Tokyo, and Niloy J. Mitra from University College, London.

Chapter 2

Background

This chapter reviews typical product design procedures widely adopted in industry to clarify this thesis's standpoint from the user's point of view. There are two approaches to the design of physically valid products. One is the combination of geometric design and physical simulation systems. The other is the use of optimization methods. We describe them in the following sections. We also depict the current use of real-time physical simulation, and briefly describe recent technical breakthroughs in production equipments that support personal fabrication.

2.1 Typical Design Workflow using Geometric Design and a Simulation System

Three-dimensional (3D) computer-aided design (CAD) systems have been widely adopted for product design in the field of engineering. Physical simulation tools for computer-aided engineering (CAE) are also commonly used to simulate the actual working conditions of a product to verify its design. With CAD and CAE tools, the user can design an object and verify it relatively easily without making an actual physical prototype. Thus, these tools play an essential role in the design of engineering products. A new design is initially created in a CAD system and then iteratively revised based on CAE analyses. In a CAD-centric process, the user manipulates the design in the CAD system, and then the geometric data are transferred to the CAE system [97]. The efficiency of design verification depends on the smooth transfer of data from the CAD system to the CAE system; it is crucial to combine the CAD and CAE environments closely and seamlessly to improve the product design [110, 42]. The various design environments of these two systems can be generally classified into two categories: stand-alone design and simulation environments (the traditional method), and a combined design and simulation environment in which the design tool and simulation tool coexist in one software package. Historically, the two environments have been kept separate, but CAE systems have been gradually integrated into CAD systems as major CAD and CAE companies merged. Many widely used CAD tools such as CATIA [37] and NX [129] have a simulation tool as a plug-in and fall into this category. Providing simulation tools inside a design package may make it more convenient for users to transfer geometric information via files. However, simulation systems integrated in a design system usually have limited capability. Furthermore, physical simulations in such combined systems are typically performed by designers, who may lack specialized knowledge in simulation, and the simulations are generally performed during design sessions and thus are limited to small-scale problems for which results can be obtained relatively quickly. Hence, separated design and simulation tools remain popular, even though the user has to manually transfer the geometric data from the CAD system to the CAE system. Many tools, such as ANSYS [7] and MSC Nastran [149], focus on simulation rather than geometric design. Such software packages often have more capability than simulation tools that are integrated into design software. They can handle more complex physics and large-scale analyses. Engineers responsible for the verification of a design, but not the creation of the design, tend to use these CAE systems. Designers send their designs to such engineers, and then must wait for the verification results, which is typically a slow process. These CAD and CAE systems are intended for trained users with specialized knowledge. Both types of systems have an excessive number of functions, and becoming proficient in their use requires many days of training and experience. Hence, it is difficult for novice users to use such tools to design physically valid objects.

2.2 Typical Workflow of Design Optimization Systems

Optimization systems typically do not make extensive use of computer graphics for simulations because simulations are mostly used to generate animations that are appealing, and aesthetic value is difficult to express in a mathematical equation. On the other hand, optimization is frequently used in engineering design because the target of optimization is much simpler: it is typically performed to increase the structure's functionality (e.g., structural soundness) while minimizing the amount of construction material needed (reduce costs). There are three main optimization methods for engineering software: the *brute-force* approach, the gradient-based approach, and the topological approach. The brute-force approach computes all of the possible combinations of the parameters and chooses the best ones. Because each parameter takes continuous values, and it is impossible to compute all of the possibilities, such systems typically divide the range of each parameter into discrete values and then run an optimization procedure over those values. Still, it is difficult to check all of the possibilities when there is a large number of parameters. For example, if we divide the range of each parameter by ten and there are five parameters, the number of total samples required is 10^5, which is not a reasonable number considering computational cost. Hence, this approach is often chosen only when the number of parameters is limited, usually from one to three. An advantage of the approach is that it entails simply changing a shape and running the simulation many times, and thus a simple batch program can perform the optimization. Many commercial software packages support this type of optimization, including DesignXplore by ANSYS [8] and Sculptor by Optimal Solutions Software [150].

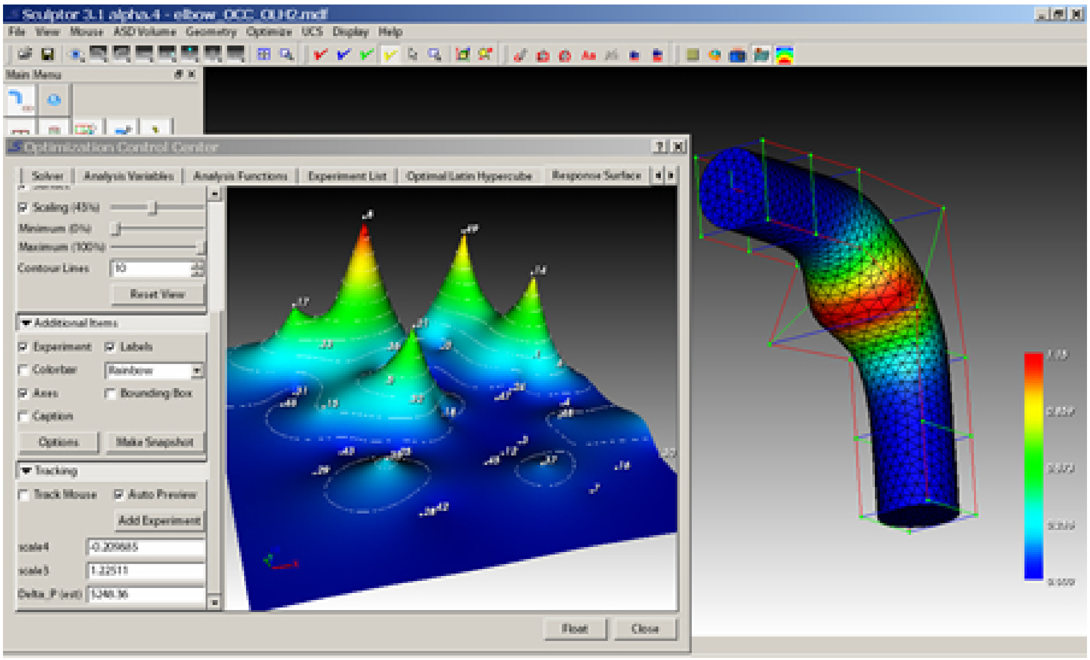


Figure 2.1: *Brute-force optimization of a flow inside an elbow pipe.*

The *gradient-based approach* computes the gradient of the evaluation function with respect to changes in its shape, and adjusts the shape parameters in the direction of the gradient (i.e., steepest descent). This approach can handle many degrees of freedom without leading to excessive computation time. However, it is limited to cases where the gradient of the evaluation function can in fact be computed. In the case of dynamic and history-dependent simulations, the gradient of the evaluation function with respect to the initial shape is difficult to compute. In addition, the gradient computation requires considerable implementation effort. Hence, software that handles this type of optimization tends to be expensive. Finally, the topological approach identifies an optimal shape from among various candidates with different topologies. This method can optimize an objective function taking into consideration a greater variety of shapes. Its main shortcomings are that the shape optimization does not consider the cost of manufacturing and the optimized shape tends to have a complex topology, making it difficult to actually manufacture. It is also difficult to consider complex evaluation functions or physical constraints with this approach. Hence, this method is commonly used only in the early stages of design to obtain inspiration and ideas that may or may not be applied to more realistic designs. Many commercial software systems, including ANSYS [7], Abacus [2], and Nastran [149], have plug-ins for this approach.

The brute-force and gradient-based approaches change the parameters of shapes

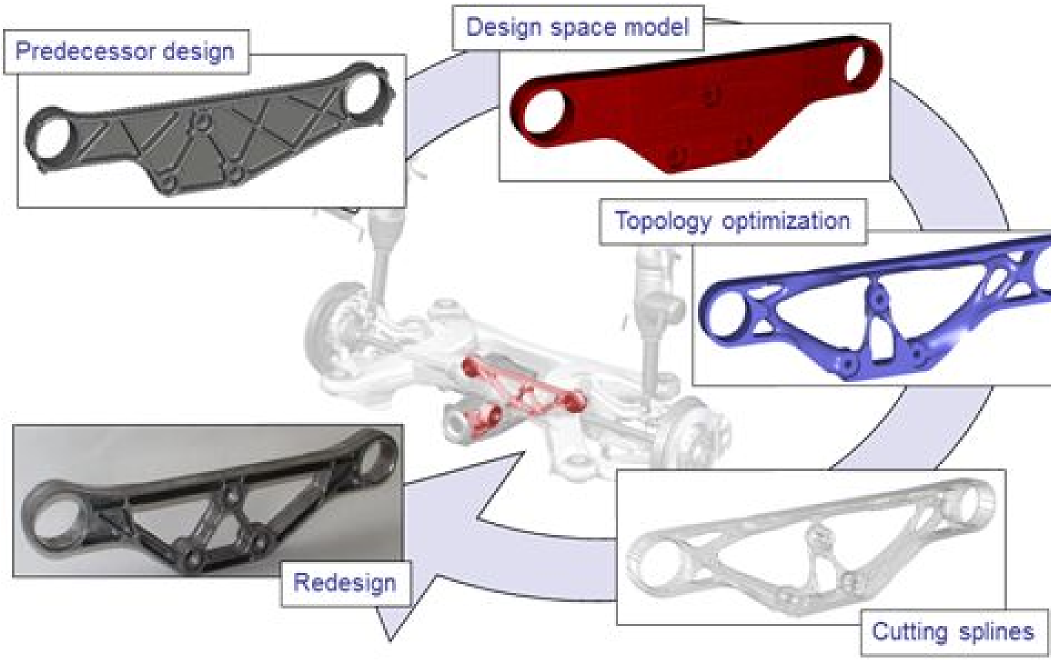


Figure 2.2: *Workflow of topological optimization.*

without changing their topology, and thus both are used in the final stages of design to optimize functionality. In contrast, the *topological approach* is used in the early stages of design, as described above. Note that all approaches run computations offline, and hence designers must wait for the optimization to finish, which can take several hours or days. No user interaction is allowed during the optimization process.

2.3 Typical Use of Real-Time Simulation

Real-time simulation allows users to interact with the simulation, and is mainly used in games and virtual reality training environments. It is used to increase the reality of a virtual world in which a game player or other user is immersed. Real-time rigid-body simulators have already been integrated into various game engines, and deformable-body simulators are becoming increasingly popular. The latter generally uses simplified mass-spring systems to simulate deformations [82]. Commercial game engines such as NVIDIA PhysX [128], Havok [72], and Bullet [101] further exploit graphics processing units (GPUs) to accelerate the simulation. Although these systems can achieve real-time performance, their computational cost is still high when generating high-quality results. Compared to finite element model (FEM)-based solutions, the quality of mass-spring systems heavily depends on the mesh structure. Other popular applications of real-time simulation include training environments using virtual reality techniques. For example, real-time deformation techniques studied in computer graphics have been applied to surgery training systems [114]. A significant trade-off exists between accuracy and interactivity in physical simulation. Many methods try to simulate deformable bodies in real-time by sacrificing accuracy using reduced or approximated methods such as mass-spring systems. Accurate real-time simulation using FEM has not been used because it is too complex and costly. Ishii et al. [83] envisioned a system in which the human and computer interact through tangible objects. They presented a design system for urban planning where real-time wind simulation was performed during the user's interactive building arrangement. However, their simulation model was simplistic. There have been many technical obstacles to providing accurate real-time simulation feedback during design. This thesis addresses the technical problem of running costly real-time FEM computation during shape design, by presenting novel algorithms and solid implementations.

2.4 Rapid Prototyping Techniques

Rapid prototyping techniques have been widely used in product design. They are mostly used by industrial designers that design a wide variety of products from car parts such as engine blocks to consumer electronic devices such as mobile phones. Current three-dimensional (3D) printing systems can print various materials such as plastics, ceramics, and rubbers, forming all kinds of shapes. Printing machines compute cross-sections of an input 3D object and then print the layers in the appropriate order. Printing accuracy is continually being improved and, today, working mechanical structures such as gears, cams, sliding joints, and rotational joints can be printed in such a way that they actually function. Along with 3D printing systems, two-dimensional (2D) fabrication machines such as laser cutters, plasma cutters, water jet cutters, wire electrical discharge machining machines, and knife cutters support sheet material cutting. Multi-axis computer numerical control (CNC) machines carve 3D objects automatically out of blocks of various materials.

Fabrication Machines for Consumers. Rapid-prototyping systems have become faster, cheaper, and better. Researchers are investigating techniques to develop fabrication machines that are compact and cheap enough to bring into homes. A group called Fab@Home developed a desktop-sized 3D printing machine called "Fabber Model I" [55] that can be constructed by consumers themselves; it costs $3,000. A company called MakerBot developed another desktop 3D printing machine called MakerBot Replicator$^{\text{TM}}$ [107] and has been selling assembled printing machines at a cost of less than $2,000. In addition, both companies have used open-source code in their designs so that consumers can improve and customize the printing machines themselves. Both groups encourage users to share digital 3D models online to inspire further improvements.

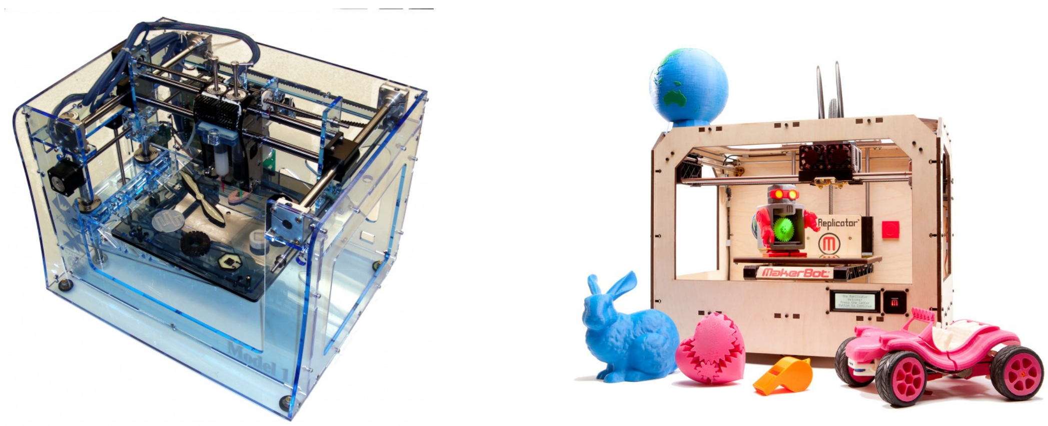


Figure 2.3: *Three-dimensional printing systems for consumers. (Left) Fabber Model I [55], (Right) MakerBot Replicator^{TM} [107].*

Chapter 3

Related Work

To provide a context for our study, this chapter reviews previous researches on physical simulation and geometric modeling. Our goal was to aid in the creation of physically valid objects by novice users, by integrating real-time physical simulations into a geometric design system. In the following sections, we focus on four topics: (i) fabrication, (ii) real-time physical simulation, (iii) design systems for novice users, and (iv) the design of physically valid objects. Then we clarify the differences between our approach and previous approaches.

3.1 Fabrication

Recently, the graphics research community has begun to show an interest in designing constructible objects using techniques common in computer graphics. For example, Mitani et al [119] presented a method for flattening a 3D object into a strip-shaped paper cutting pattern. Influenced by this work, many researchers have attempted to fabricate 3D shapes using common materials including paper [75, 112]. Li et al. presented a system for designing pop-up cards [99, 100]. In other research, input objects have been converted into constructible 3D puzzles [104] including burr puzzles [174]. In Beady [177] , a 3D shape was realized using beads, and the authors computed the path of threads going through the beads. Lau et al. presented a system for converting a 3D furniture model into fabricable parts and joints [96]. Optimization strategies have been proposed for handling lighting effects such as shadowing, addressing problems such as voxel selection and placement to create multiple shadows [120] or relief that causes shading under specific lighting conditions [4]. Techniques to generate a 3D-printing shape with a desired subsurface scattering property were presented in [73, 171]. Some studies have also addressed deformable bodies. For example, [146] studied the transition of deflated rubber balloons into inflated target shapes, and [24] used a combination of deformable materials to achieve certain deformable properties. Mori et al. proposed a system called Plushie [121] that designs a 3D plush toy that is stitched from 2D fabrics.

.

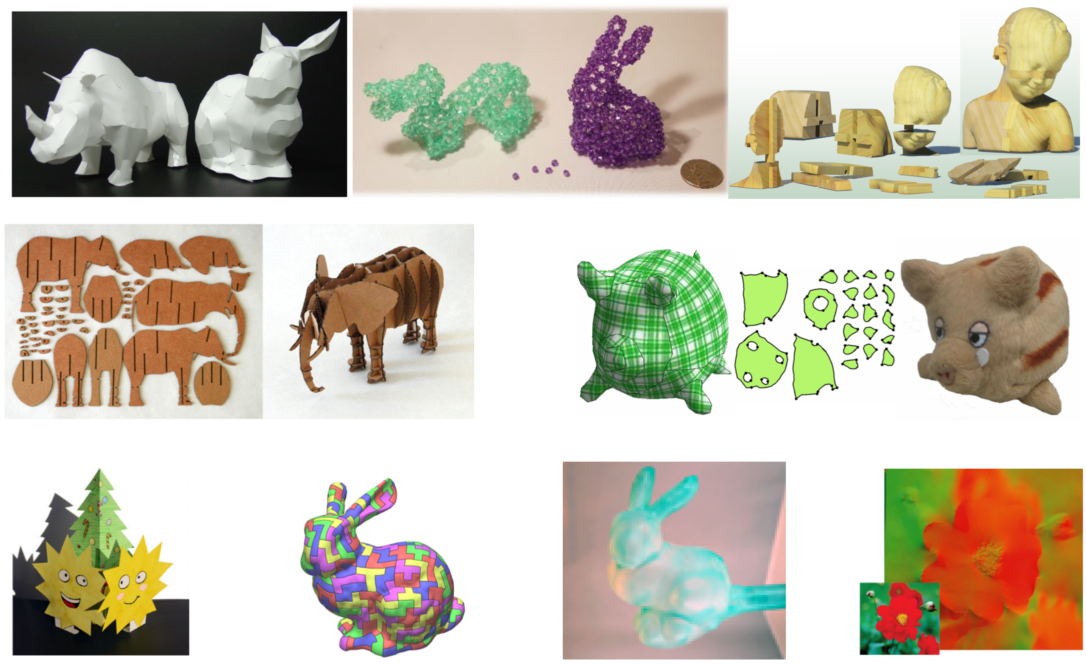


Figure 3.1: *Some results of fabrication researches. From top left to bottom right, papercraft [119], beadwork [177], burr puzzle [174], assembly of paper with slits [75], plush toy [121], pop-up card [99], 3D polynomial puzzle [104], and rubber balloon [146].*

Discussion

Most fabrication research has focused on how to achieve fabricable output that is as close to a user's input shape as possible. Making things fabricable can be viewed as imposing constraint on a shape. For example, realizing a 3D surface with paper involves imposing a developable constraint on the surface. The difficulty of fabrication depends on the degree of constraint. The input and output are very similar under lesser constraints and very different under higher constraints. Thus, users have to find a balance. In addition, most fabrication research has focused on one-way optimization, that is, computing fabricable objects from a user's input shapes rather than facilitating communication and interactions between users and the system.

3.2 Real-Time Simulation

Since Terzopoulos [155] introduced simulations of deformable objects into computer graphics, various approaches for making physical simulations interactive have been studied. One approach is to leverage pre-computations. For example, James and Pai [85] performed an interactive physical simulation of deformable objects by pre-computing the flow in the high-dimensional phase space of deformable animations. James also proposed modal analyses of dynamic elastic models for real-time deformation simulations [86]. Second-order deformation modes have been used to compute nonlinear deformation in real-time [13]. Another approach is to use a co-rotational framework to approximate geometric non-linearity to achieve large deformations quickly and stably [122]. Domain decomposition techniques have been used to generate real-time deformation animations of tree-like structures [16]. Parker simulated highly nonlinear phenomena such as fractures in real-time in game applications [133]. Chentanez simulated coupling deformation between a rod and solid deformable material in real-time to facilitate needle insertion in a surgery simulator [41]. For more studies on real-time deformation simulation for surgery training, see the detailed review of [114]. Collision detection between deformable objects is usually computationally costly, and often presents a bottleneck in real-time simulations. Barbič et al. [14] proposed a practical method for detecting collisions between rigid bodies, and Allard [5] demonstrated the use of real-time collision detection between deformable objects using a technique called layered depth imaging. A number of researchers have attempted to simulate clothing deformations quickly for real-time character animation (for an overview, see [125, 44]. Clothing deformation is usually associated with character pose changes. Hence, pre-computation approaches (e.g., the regression approach or database approach) are often used. De Aguiar et al. [48] presented a learning approach that correlated character pose and dynamic cloth deformation. Wang et al. [170] presented a database approach for transferring pre-computed high-resolution wrinkles to a low-resolution cloth simulation. Finally, there have also been many studies on real-time fluid simulation. The Stable Fluid System [151] computes fluid-like animation stably in real-time. A modal reduction technique for real-time fluids was presented in [158]. Wicke et al. presented a modular-based fluid animation synthesis [173]. Batty et al. [20] presented a real-time coupling simulation of fluids and rigid bodies using various techniques.

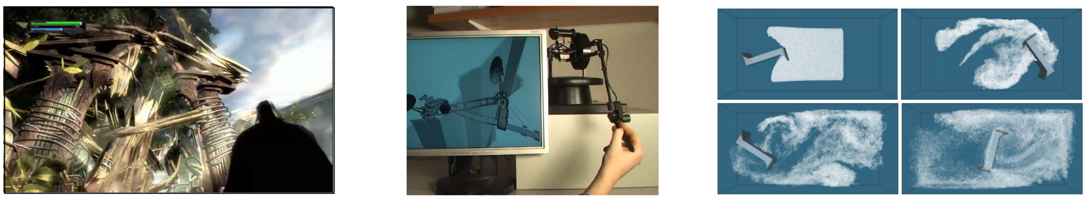


Figure 3.2: *Results of real-time simulations. Left, real-time simulation of a deformable object with fractures in a game environment [133]. Middle, real-time deformation for haptic rendering [13]. Right, real-time simulation of the interaction between a fluid and a solid [20].*

Discussion

All of the above studies implemented a fixed initial rest shape. Users can input external forces into a continuously running simulation, but the rest shapes of the materials cannot be changed. It is difficult to change rest shapes in real-time when a simulation uses pre-computations, because if a rest shape changes, all the pre-computations associated with it become useless.

3.3 Shape Optimization

Rest Shape Optimization for Goal Deformation

In typical geometric design systems, a user designs a rest shape without any external forces. In subsequent physical simulations, the rest shapes may be deformed in a way that the designer does not want. Hence, many researchers have studied optimization methods for computing rest shapes that result in user-specified goal deformations after applying a simulation. Alexandre et al. [51] presented a rest shape optimization system that determines the rest shape of a 2D rod that conforms to a 2D sketch drawn by the user. Their system iteratively finds material parameters that satisfy the task as well as possible while keeping the rod as flexible as possible. Twigg et al. proposed a similar method for 3D objects [162].

Mélina et al. [146] presented a shape optimization method for rubber balloons made of thin latex membranes that were inflated to a specific target shape. Bickel et al. [24] presented a method for optimizing combinations of materials to achieve target nonlinear deformation behavior under a specific load.

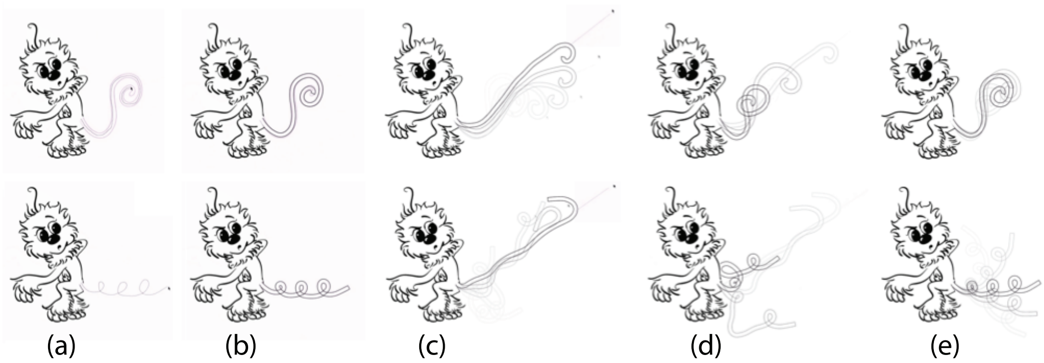


Figure 3.3: *Examples of an optimized rest shape presented by Derouet-Jourdan [51]. (a) The user sketches a tail of a character, and (b) the optimization procedure determines the rest shape of a two-dimensional rod that would deform into the user's sketched shape under gravity. (c-e) The elastic rod allows the user to interact with the sketch.*

Optimization of Physics-Based Animation

Small changes to initial parameters can lead to large changes in the final configurations of a physics-based animation, making it difficult to control the animation.

Popović et al. [135] presented an interface through which users interact directly with a simulation to produce a desired result. The so-called many-worlds browsing system [160] allows users to see a variety of physical simulation results to identify the desired animation. Our research applies a similar principle to design physical objects with FEM simulation. In another interesting formulation, Twigg and James [161] introduced a backward-step simulation to generate animations of rigid bodies with desired configurations. Earlier, Chenney and Forsyth [40] extended traditional simulation models to include plausible sources of uncertainty, and used a Markov Chain Monte Carlo algorithm to sample multiple animations while satisfying a set of constraints to ensure the physical plausibility of the animations. Beyond rigid bodies, the optimization of animation has been studied in several other domains. Barbič et al. proposed a method for extending users' key-frame control of animation to deformable objects [15, 12]. Animations of smoke, optimized based on user-specified constraints using key-frames, have also been demonstrated [159, 113, 56].

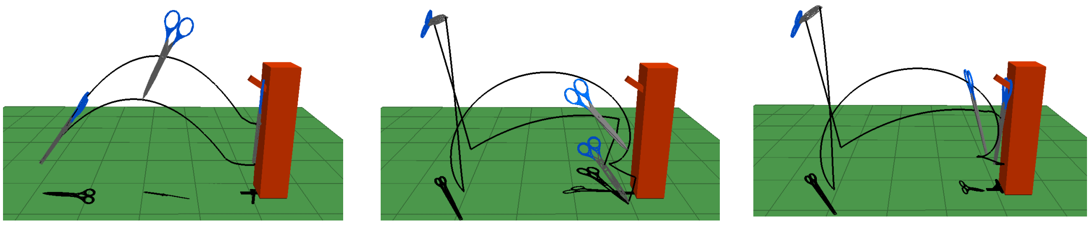


Figure 3.4: *Optimization of animation presented by Popović [135]. The system develops an animation that achieves a specific goal specified by the user (e.g., hanging scissors on a hook). The user can interactively change the initial configuration of the rigid body.*

Architectural Design Optimization

In the context of architectural design, researchers have worked on optimization methods for determining designs that can be constructed economically while minimizing changes to user input. To this end, tessellations of free-form surfaces have been used to maximize repetitions across molds or triangular patches [53, 145]. The geometries of free-form surfaces can be optimized in various ways to support construction; examples include the use of single curved panels [137], multi-layer structures [136], and/or conical meshes [103]. Yan et al. [175] presented a system for designing paneled structures that allows users to interactively manipulate parameters with a bound computed by the system. Another previous study optimized floor plans inside buildings considering various constraints from simple user input [115]. Several studies have also investigated how to arrange furniture inside a room while also increasing functional considerations such as accessibility [116, 176]. Some researchers have attempted to optimize physical properties. For example, Smith et al. [148] modeled truss structures by structural optimization to generate visually plausible shapes used in computer graphics. The system minimized the evaluation function, G, which was chosen as the total mass in the system, with respect to joint positions, $\vec{q}$, as follows:

$$\begin{aligned}
\min \quad & G(\vec{q}), && (3.1)\\
s.t. \quad & \vec{F}_i(\vec{q}) = 0 \quad i = 1, \ldots, N_J, && (3.2)\\
& \|\lambda_j\| \leq \lambda_{max} \quad j = 1, \ldots, N_B, && (3.3)\\
& \left\|\vec{l}_j\right\| \geq l_{min} \quad j = 1, \ldots, N_B, && (3.4)
\end{aligned}$$

where $F_i = 0$ denotes the equilibrium of joints i $(i = 1, \ldots, N_J)$, λ_j is an axial force, and $\left\|\vec{l}_j\right\|$ is the length of each beam j $(j = 1, \ldots, N_B)$. The user specifies the initial positions of the joints as well as the constraints on the fixed joint positions. The system adds many beams between the joints as an initial attempt to connect them, and then through optimization, removes useless joints that do not support the structure and updates the joint positions by minimizing the evaluation function G.

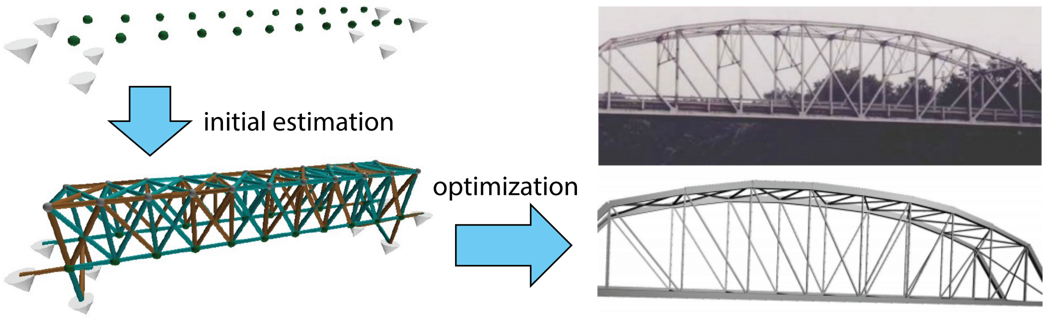


Figure 3.5: *Example of an optimized truss structure [148]. The user specifies the initial joint positions as input (upper left). Then the system generates many beams as an initial attempt to connect them (lower left). An optimized truss structure (lower right) that is similar to truss structures in the real world (upper right).*

Whiting et al. [172] optimized free variables in the context of procedural modeling with respect to structural feasibility by ensuring a non-negative force between brick elements, as follows:

$$\begin{aligned}
y(\theta) = \quad & \min_{\mathbf{f}} \quad \sum_{i=0}^{n} \left(f_n^{i-}\right)^2, && (3.5)\\
& s.t. \quad \mathbf{A}_{eq} \cdot \mathbf{f} = -\mathbf{w}, && (3.6)\\
& \mathbf{A}_{fr} \cdot \mathbf{f} \leq 0, && (3.7)\\
& f_n^{i+}, f_n^{i-} \leq 0 \quad \forall i, && (3.8)
\end{aligned}$$

where θ is a parameter in the design, $\mathbf{A}_{eq}$ is a coefficient matrix that distributes the external force $\mathbf{w}$ to a contact force $\mathbf{f}$, and f_n^{i+} and f_n^{i-} are the positive and negative components, respectively, of each component at each contact point $f_n^i = f_n^{i+} - f_n^{i-}$. The system optimizes the parameter θ such that the entire contact force becomes positive.

Figure 3.6: *Structural optimization of masonry architecture studied by Whiting et al. [172]. Through the optimization process, the system produces a visually plausible model that can be used to simulate collapsing.*

Discussion

These methods are useful, but they are not integrated with the design phase and they do not consider any physical durability constraints of shapes. These approaches provide final *optimized* shapes, which is not useful in the initial exploratory stages of design. In contrast, our system introduces force-space analysis to investigate the effects of geometric changes on physical validity; we use the results to expose the valid and useful parts of the shape space as suggestion modes. Automatic methods present many practical difficulties, such as explicitly specifying constraints and parameter spaces that are too large. The interactive approach offers the advantage of allowing users to use their own preferences and judgment during the design process while considering less tangible factors such as aesthetics.

3.4 Geometric Design System for Novice Users

Interactive Modeling Systems

Funkhouser et al. [64] proposed a data-driven modeling system in which users provide a conceptual design using sketches, and the system suggests plausible geometric realizations by searching through a database of 3D models. In the same spirit, several techniques that support modeling-by-example style-shape authoring have been proposed [6, 166], as have techniques for interactive shape manipulation considering global geometric features. Kraevoy et al. [93] proposed a technique for resizing 3D shapes while preserving detailed features by changing scaling uniformly. Wires [65] demonstrated that direct preservation of inter- and intra-part relationships using junction curves is effective for manipulating man-made models. Bokeloh et al. [28] proposed a system that analyzes input shapes to extract a high-level shape grammar that is utilized in procedural modeling. The SKETCH system [178] allows users to create and arrange 3D objects using simple gestures. One influential sketch-based model is Teddy [79], which allows users to intuitively model a 3D shape by sketching. The user draws a 2D outline shape, and then the system inflates the shape into three dimensions. Teddy has inspired several other modeling systems, such as FiberMesh [124], which allows users to draw curves on a 3D inflated shape and modify the polygon by dragging the curves. Olga et al. [87] proposed a system that generates 3D shapes from 2D sketches of T-junctions or cusps. In another approach, Yotam et al. [68] presented a 3D modeling system where the user inputs primitives and annotations, such as symmetry and equal length constraints, onto 2D images, and the system reconstructs the 3D shape from the user-specified information. Alec et al. [138] proposed sketch-based 3D modeling from 2D silhouettes from different view angles.

Suggestive Modeling

Advances in geometric modeling have resulted in well-established CAD modeling tools. Exploratory design, however, remains challenging. This is mainly because mapping a partially formed design concept to a final 3D shape is ambiguous. Hence, researchers have proposed various frameworks for identifying possible shapes to inspire and guide users. Based on user-specified geometric relationships across 3D components, Igarashi et al. [78] generated a gallery of possible modeling operations to facilitate quick and intuitive modeling. Inspired by modeling-by-example studies, Chaudhuri and Koltun [39] proposed a data-driven system for computing and presenting components that can be added to the current design shape. Later, they extended the idea to a probabilistic suggestion system for part-level assembly-based 3D modeling [38]. The Insitu system [132] provides a spatial context by fusing data from multiple sources and combining them with image bill-boarding to provide lightweight 3D environments for professional conceptual designs. In the context of appearance modeling, Kerr et al. [89] performed a user study to test the effectiveness of suggestive interfaces. They concluded that such an interface was well suited for artistic exploration, even for novice users, but they also stated that the interactivity of such a system is critical.

Interactive Shape Exploration

Immediate and meaningful feedback is essential in any design setting, especially in artistic exploration (see also [89]). Although such design spaces are often high dimensional, only low-dimensional subspaces are generally useful for intuitive exploration. In data-driven settings, researchers have extracted low-dimensional embeddings (e.g., using a mixture of Gaussian models) of desirable design spaces for appearance [144] and for geometric modeling [153]. Recently, Ovsjanikov et al. [131] studied variation patterns in descriptor spaces to extract low-dimensional deformation models on a representative template for the exploration and navigation of collections of 3D models. In a related attempt, Yang et al. [175] proposed a geometric framework for identifying constrained modeling spaces where appropriate geometric properties (e.g., planarity of quad faces) are preserved in the course of deformations and edits.

Discussion

Many researchers have attempted to develop intuitive geometric modeling interfaces for novice users. However, few have explored simple geometric modeling interfaces for *physically valid* objects; most research to date has focused on geometry, largely ignoring physical validity considerations.

3.5 Numerical Simulation for Musical Instruments, Clothing, and Furniture Evaluation

In this section, we describe several researches that simulate physical property of functional objects. We choose three targets, musical instruments, clothing, and furniture, which correspond to the targets of actual manufacturing in Section 4, Section 5, and Section 6. These designs have been almost always based on practical experience from traditions in handicraft manufacturing. However, recent progress in simulation techniques made it possible to evaluate physical validity of these objects before the manufacturing.

Simulation of the musical instruments is an establish field of the research and there have been a number of studies on the use of FEM eigenanalysis for predicting the tones of various musical instruments. Bécache et al. [21] performed simulation of acoustic guitar where the soundboard is modeled as a thin-shell and the guitar's acoustic transfer is computed using a time-domain fluid structure coupling techniques (see Figure 3.7). The sound of piano is mainly determined by the vibration of the soundboard. Several researches describe eigen analyses of the piano's soundboard [91, 88, 3]. In the similar manner, the simulation of the sound based on the eigen analysis is done for xylophone [33] and for timpani [98]. Shoofs et al. [142] used FEM eigenanalysis and optimization techniques to determine the optimal shape of carillon bells which have desired overtones with a harmony.

The simulation of cloth, and more generally thin-shells, is a widely investigated area [34, 11, 69]. Mass-spring models generally enable fast and simple computational models for rapid cloth simulation [117, 43] providing responsiveness sacrificing accuracy and stability. Recent methods have also generated real-time wrinkle synthesis [139], as well as more generally, real-time, data-driven reduced cloth modeling [57]. Using these clothing simulation techniques, several researches envision virtual clothing design enviroments [167, 60]. Some commercial software packages

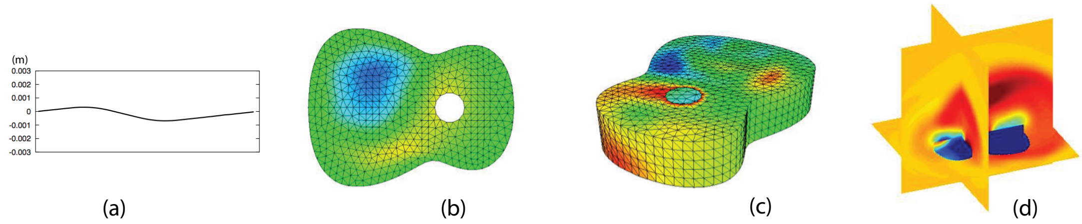


Figure 3.7: *An acoustic simulation of a guitar described in [21]. A one dimensional vibration of a string (a) induces guitar's soundboard deformation (b) which causes pressure distribution on the surface of the guitar (c) and radiation of acoustic wave (d).*

such as OptiTex [130] have already incorporated clothing simulation into the geometric clothing pattern design system.

Furniture simulations also have been performed in many research to analyze their structure under various load conditions. Since there is a well-written comprehensive summery of the finite element analysis on wood furniture [106], here we describe only the outline of research interests. Mechanical properties of fiber materials such as wood have been studied for many years [27, 76]. Stress distribution under non-static loading is studied for making furniture more reliable. Paoliello et al. [36] studied loading on a Eucalyptus wood chair during mens' sitting activities. Mustafa et al.[123] evaluated durability of a wooden chair in cases of free fall using finite element method based impact analysis. Dynamic simulations that reproduce durability tests prescribed in the International Organization for Standardization (ISO) [84] were performed for laminated bamboo structures [95]. Many researches study about mechanical property of various types of joints that are used to connect planks. Load carrying capacities of bolted timber joints were studied in [46, 105]. Properties of the adhesive joints were mathematically modeled in [45, 147]. Nicholls and Crisan [126] studied on the stress-strain state in corner joints and box-type furniture using finite element analysis.

Discussion

Simulation methodologies have been developed for specific targets and it become possible to evaluate accurately physical validity of the functional objects. However, these simulations take considerable time to perform and have been used exclusively as offline evaluation tools and have not been directly integrated into geometry editing as in our system.

3.6 Physics Simulation Inside Interactive Design

Several systems that design physical objects have been proposed. Masry and Lipson [111] developed a sketch-based 3D modeling interface capable of FEM analysis. Figure 3.8 shows flowchart of this system. The user sketches a 2D outline of 3D strokes, and then the system reconstructs the 3D shape via optimization considering various constraints such as the regularity of 3D edges. The system runs the FEM simulation and presents the results to the user. The user observes the simulation and re-sketches the shape according to the observations.

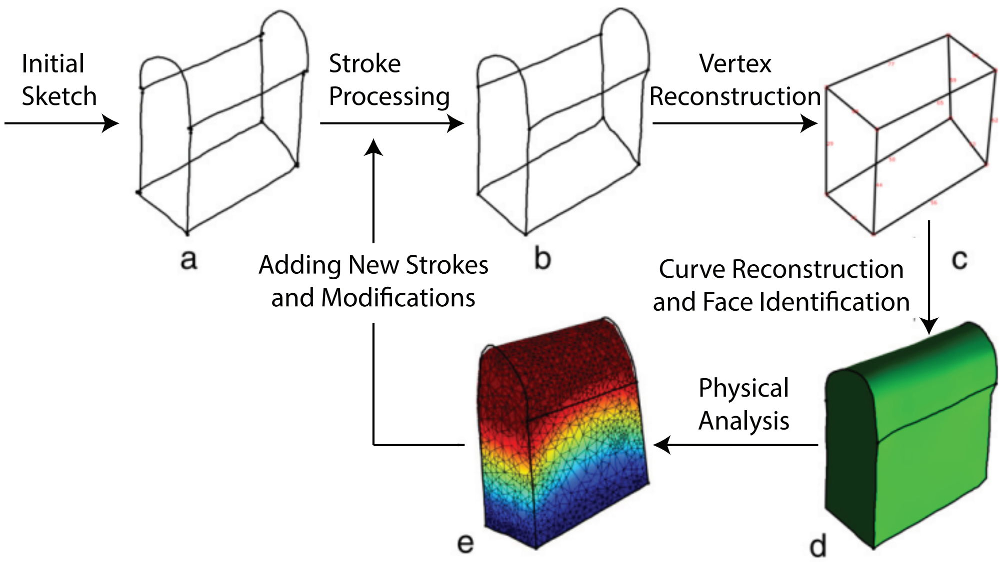


Figure 3.8: *Sketch-based engineering design system. The user sketches an outline of a shape and the system identifies the corresponding 3D shape. Then a finite element analysis is performed on the generated 3D shape and gives the user feedback on the structural soundness for further editing.*

Whiting et al. [172] proposed an interactive model system for masonry structures. A structure is represented with a few design parameters using a combination of design template substructure parameters. Masonry structures have a physical constraint in that each block is connected to others via non-negative contact forces. To satisfy these constraints, the system runs an optimization process. Different from previous offline optimization approaches, such as that proposed by [148], this optimization is interactive and users can investigate structural optimization during editing. Figure 3.9 shows a flowchart of this system.

As discussed above, Plushie [121] is a modeling system for stuffed animals that relies on interactive simulations. The system is equipped with a Teddy-style [79] 3D modeling process where users can change the shape of a pattern intuitively. Figure 3.10 shows the workflow of the system. When a user draws a stroke, the system runs an optimization to create a matching 3D outline of that stroke. When the user

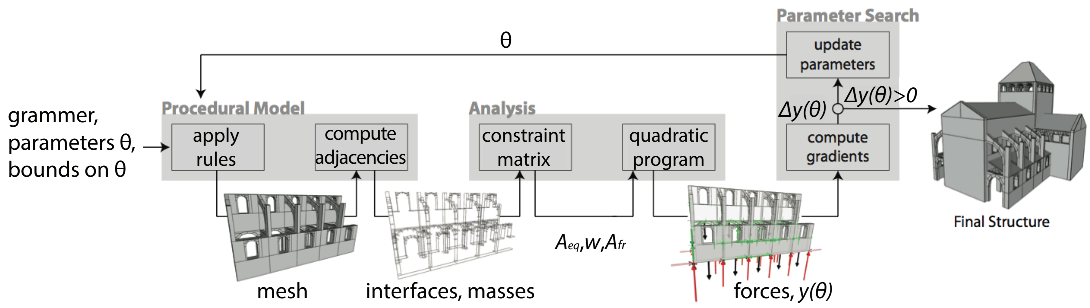


Figure 3.9: *Flowchart of the masonry structure design system [172].*

edits a pattern, the system runs a real-time simulation to predict the 3D shape of the plush toy. By running the simulation and optimization concurrently during modeling, the output is always realizable, and thus the user does not get stuck modeling unrealizable products.

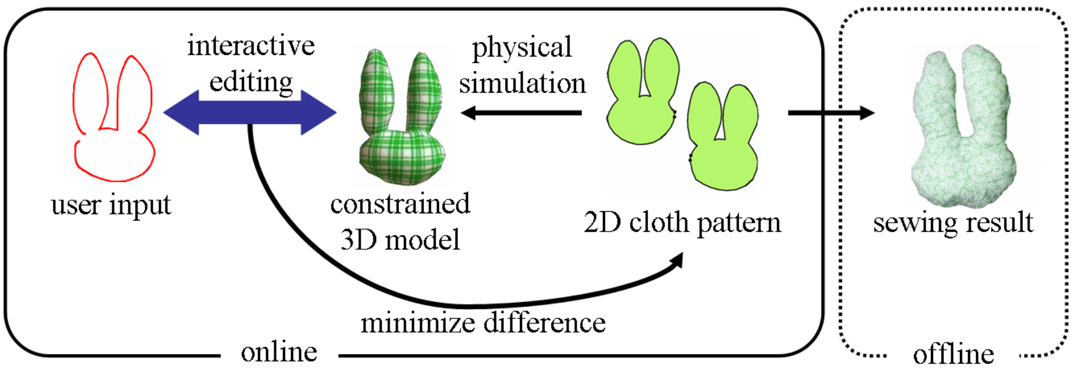


Figure 3.10: *Workflow of the plush-toy design system presented in [121].*

In the field of engineering, some researchers have begun to realize the importance of using simulation in the early stages of a design. For example, Nishigaki et al. [127] proposed the concept of first-order analysis (FOA), which performs approximate finite element simulations using structural elements with a low degree of freedom to explore design principles in the early stages of design.

Discussion

Modeling systems with physical simulation capability, such as Masry and Lipson's sketch-based modeling system, allow users to easily test designed shapes. However, such systems are not very different from traditional CAD and CAE systems in that the analysis is performed after some of the design processes have already occurred. Optimization during interactive shape editing, which Whiting's masonry design system achieves, is a promising approach. By performing optimization, a designed shape is kept physically valid and users can get feedback about how the optimized shape changes during editing. However, there still remain problems originating from the optimization. For example, the system may propose one optimized shape, but may not show changes as the user varies the design. The Plushie system is innovative in that it ensures that the constructed model is physically correct using real-time simulation. However, the focus of this system is mainly on the user interface, and not on the simulation. The simulation only supports low-resolution meshes and simple physical cloth models. The FOA concept greatly reduces the trial-and-error associated with undesirable simulation results in the later stages of design. However, simulations that use FOA are performed after the conceptual design is finished and do not provide suggestions for improving the design.

Chapter 4

FEM Framework for Changing Rest-Shape

This chapter shows a framework intended to compute FEM simulation response during the editing of rest-shape. FEM is one of the most common physics simulation techniques used in the product design. However, the typical FEM framework is difficult to compute response with respect to the rest-shape change quickly; if the rest-shape changes, the system runs simulation from scratch generating the mesh and building data structure of the linear system solver. We propose an acceleration technique of FEM simulation with respect to the rest-shape change by reusing the redundant data which undergoes no or only slight change. We investigate the procedure of FEM and find out which data can be reused. In our approach, when the rest-shape changes, the mesh is deformed to fit into the changed geometry, which reduces the cost of mesh generation. The FEM internal data structure is also reused according to the level of the mesh change.

This approach is quite general and can be applied for many types of FEM problems, including nonlinear problem, non-static problem, and eigenvalue problem. To demonstrate the generality of our algorithm, we implemented interactive systems for several problems including vibration, fluid, thermal fluid, and standing wave. We performed two informal user-studies; one is custom metallophone design, and the other is bridge design. These studies show that novice users can design objects with complex physical constraints using the real-time feedback from the FEM simulation.

4.1 Introduction

Interactive numerical simulations can be a powerful tool for assisting the design of various items that satisfy specific physical requirements as described. However, most numerical simulation methods today, generally referred to as computer-aided engineering (CAE) tools, are used for the off-line verification of a given design. They are used to reject designs that fail to satisfy the requirements, but are not usually used to explore a better design. Real-time simulation is emerging, but typical applications are the simulation of deformation in animation [118] and virtual training [41]. Real-time numerical simulation is not widely used as a tool for designing physical items.

This chapter introduces our efforts at integrating a numerical simulation method into geometric modeling as described in Sutherland's [152] vision. Our system runs a finite-element method (FEM) simulation in real time that responds to dynamic user input during geometric editing. Unlike standard real-time FEMs for deformation that are applied to a single given initial geometry, our method continuously updates the simulation results responding to the initial geometry being modified. Real-time feedback during editing can provide guiding principles for better design and help the user approach a satisfactory design while avoiding many trials-and-errors experiments necessary with an off-line simulation. Responsive feedback is also useful for educational purposes to learn the relationship between the shape and physical behavior.

The technical contribution of this chapter is the way in which we modify the traditional FEM framework to make it responsive, that is, to make it efficiently update the computation result responding to the continuously changing initial boundary geometry. The key observation is that the geometry only gradually changes when the user modifies the boundary by direct manipulation (i.e., by dragging the mouse). In this case, the mesh only slightly changes and then most matrix computations can be reused to accelerate the computation. The important questions are which computations to reuse and when. To answer these questions, we decompose the computation into multiple reusable components and perform the appropriate amount of recomputation by monitoring the dynamic user input. When the modification is small, the system only slightly updates the mesh, and most matrix computations are reused. For the large modification, the system gradually makes larger changes to the mesh and updates more matrix computations to maintain accuracy. We show that this method significantly improves the performance compared simply to running a monolithic FEM each time.

We present several example applications to explain our concept including structure vibration, structural analysis, fluid, and thermal fluid. We also performed two informal user studies to show the effectiveness of our approach. One was to ask a professional artist to design a customized metallophone using the responsive FEM analysis. The other was to ask nonprofessional test users to design a bridge with and without responsive FEM. Although our current implementation is limited to 2D problems with simplex first-order elements, the basic concept of responsive FEM is independent of dimensionality and element types.

Our contributions are summarized as follows:

- We propose a responsive FEM framework in which the simulation result is continuously presented to the user during geometric editing.
- We introduce a solid implementation to support the vision. It incrementally updates the FEM data structure to avoid redundant computation.
- We present several example applications, each of which is innovative and useful

in itself.

- We conducted two informal user studies to demonstrate the effectiveness of our approach.

4.2 Multi-Level Data Reuse for Responsive FEM

This section describes how to make the FEM framework responsive, that is, to provide immediate feedback during geometric editing. We achieve this by maximizing the reuse of intermediate computation results and carefully scheduling the computation pipeline to provide the best user experience. We first briefly describe the basic FEM framework as the basis of our algorithm. We then describe our proposed method to make FEM responsive, followed by detailed description of our current implementation.

4.2.1 FEM Background

FEM finds an approximate solution of partial differential equations by spatially discretizing the field. The system first constructs a mesh inside of the boundary geometry and then solves a linear system $A\boldsymbol{x} = \boldsymbol{b}$ that is defined by the relationships among nodes (note that for nonlinear problems we need to solve such linear systems iteratively). Since the matrix A is sparse, it is compactly represented by the combination of the nonzero pattern A_p that represents the location of nonzero elements, and the value list A_v that represents the values at the nonzero elements. Iterative methods are commonly used to solve sparse linear systems and their performance is often improved using a preconditioner. A preconditioner B is used to approximate the inverse of A which is not necessarily sparse. B is usually represented as a sparse matrix with its nonzero pattern B_p and the value list of the nonzero elements B_v.

These data (A_v, A_p, B_v, and B_p) must be constructed before actually solving the system. The construction of the mesh and the matrices can be considered as a precomputation. When solving a linear problem, the system runs the entire precomputation only once. The system finds a solution without changing the matrices and reuses them multiple times to solve a time-varying problem. In contrast, the system needs to solve the problem iteratively updating A_v and B_v each time to solve a nonlinear system.

Traditional FEM frameworks run the reconstruction of mesh and matrix computations all at once for every change of the geometry. When the user applies the same analysis to even a slightly modified geometry, the system discards the result of all precomputations and starts construction of the mesh and matrices from scratch. This is a waste of time because most of the computations are redundant and can be reused. The next section describes how we modify the FEM computation process to achieve this goal.

4.2.2 Our Approach: Multi-Level Reuse

In our system, the user modifies the boundary geometry by dragging vertices, edges, or regions, and the system continuously runs FEM analysis on the domain. Note that in the case of structural analysis, the user modifies the rest-shape, not the deformed shape emerging as a result of simulation. The challenge is to provide immediate feedback to the user while maintaining a certain level of accuracy. Making a system *responsive* is not the same thing as simply making the system *fast*. One needs to be careful in distributing the computation corresponding to the degree of change in data caused by the user to maximize the speed–accuracy trade-off. We achieve this goal by reusing intermediate computation results instead of recomputing everything every time the boundary geometry is modified.

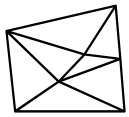
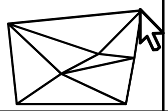
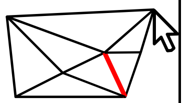
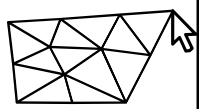
The basic concept is as follows. When the geometric modification is small, we can reuse most of the previous intermediate computation results to obtain an accurate result. If the accumulated geometric modification becomes too large, then we stop reusing previous results and run the costly computation to maintain accuracy. To implement this concept in a FEM framework, we divide the computation into multiple stages and choose the appropriate amount of re-computation depending on the current situation.

As we described in Section 4.2.1, the FEM main computation is divided into mesh construction and matrix computation. When the boundary geometry is modified, then the mesh and matrices need to be recomputed. We define three levels of re-computation and choose the appropriate one to balance speed and accuracy (Table 4.1). Continuous update of a mesh during simulation is already used to solve problems that involve geometry changes such as fluid–structure interaction based on Arbitrary Lagrangian Eulerian methods. However, such off-line methods do not selectively apply different update procedures responding to the user input as in our method.

Level 1. When the modification of the boundary geometry is small, we only change the position of the mesh nodes (relocation). This does not change the topology of the mesh. Therefore, we only need to update the value list of the linear system (A_v), while reusing all the other data (A_p, B_v, and B_p). We can also reuse the FEM solution in the previous configuration. Since the nodes are moved only slightly, the solution (field values at the nodes) does not change very much. We therefore reuse it as an initial guess in running an iterative solver; this is faster than starting from an arbitrary guess.

Level 2. When the modification of the geometry becomes larger, node relocation is not sufficient to eliminate distortions in the mesh and we change the topology of the mesh locally to improve the mesh quality (reconnecting). In this case, we need to update the nonzero pattern A_p as well as the value list A_v. However, we can still reuse B_v and B_p because nodes are not added or deleted, and they are only slightly moved. Reuse of the preconditioner is a known technique, but it is used mainly for

Table 4.1: *Multilevel reuse. A check mark indicates that the data can be reused. A blank means that the data needs to be recomputed.*

✓ = reusable	User operation	Idle	Dragging		
			Level 1 (Relocation) 	Level 2 (Reconnecting) 	Level 3 (Reconstruction) 
Coefficient matrix A	Value list A_v	✓			
	Non-zero pattern A_p	✓	✓		
Pre-conditioner matrix B	Value list B_v	✓	✓	✓	
	Non-zero pattern B_p	✓	✓	✓	

solving nonlinear problems. The solution can also be reused as the initial guess in the iterative solver as in the Level 1 case.

Level 3. Even reconnecting is not sufficient when the modification of the geometry is significantly large. In this case, we stop the incremental update of the mesh and reconstruct the entire mesh from scratch (reconstruction). This might sound too radical, but a global reconstruction is often faster and yields a better mesh than local optimization with node insertion and deletion when the distortion has accumulated or the boundary geometry is too different from the current boundary. In this case, we recompute all data: A_v, A_p, B_v, and B_p. In addition, we cannot reuse the previous solution because the old nodes are completely replaced by new ones. Therefore, we need to start with a new initial guess.

We reuse FEM data to maximize the responsiveness of the analysis by considering the cost required for each level of recomputation; we try to rely mostly on the lightweight Level 1 recomputation while performing the expensive Level 3 recomputation only when necessary. The basic concept described above applies to both linear and nonlinear problems. However, the details are slightly different in nonlinear cases. Specifically, the value lists A_v and B_v need to be updated each time when solving a nonlinear system, so we cannot reuse them. However, we can still reuse the nonzero patterns A_p and B_p, which significantly contributes to improving the performance.

4.2.3 Implementation Details

The reuse of FEM data is divided roughly into the reuse of the mesh and the matrix computations. The multi-level reuse first determines what part of the mesh structure to reuse and then uses this to decide what part of the matrix computation to reuse. The system changes the mesh to a limited extent of element destortion. When the user edits the boundary geometry, the system first relocates the nodes to minimize mesh distortion. If the system detects an inverted element after the relocation, the system pushes the nodes back to the previous positions and applies reconnecting. If reconnecting does not occur, it means that reconnecting does not improve the mesh quality and the system reconstructs the entire mesh. If no inverted element is detected after relocation, the system checks for the existence of distorted elements. If distorted elements exist, the system applies reconnecting.

We use a simple mass-spring system for the node relocation. The rest length of spring is zero and we solve equilibrium iteratively. Since the nodes move only slightly each time, the computation converges quickly. An even number of iterations is desirable to avoid oscillation; we currently perform two. The distortion metric is based on the ratio of an inscribed circle and the maximal edge length. We apply edge swapping for reconnecting. The criterion of an edge to be swapped is whether the edge violates the Delaunay condition. Mesh reconstruction is based on Delaunay triangulation and local optimization.

The conjugate gradient method is used for solving symmetric matrices, while the Bi-CGSTAB method is used for solving asymmetric matrices. We improve the convergence of these iterative methods by using the preconditioner based on the incomplete LU factorization with level of fill-in (ILU(k)). The ILU factorization method computes a sparse matrix B that approximates the inverse of a sparse matrix A (note that the exact inverse of A is not sparse in general). The method takes an integer called 'level of fill-in' as a parameter, which specifies the level of the approximation. It affects both the improvement of the convergence and the cost of the factorization; the higher the level, the more closely B approximates the inverse of A leading to a faster convergence, while requiring more computations for the factorization. A preconditioner with a high level fill-in benefits more from our multi-level reuse scheme, because the number of the preconditioner recomputation is much reduced in our method. However, the best level of fill-in is heavily dependent on the target problems, and we experimentally chose appropriate ones for each application (e.g., we used the three level of fill-in for the vibration analysis and the cantilever deformation examples, while we used the zero level of fill-in for the fluid and the thermal fluid examples in Section 4.3).

4.2.4 Performance

We demonstrate the effectiveness of our multi-level reuse described above through an example modeling sequence shown in Figure 4.1. Table 4.2 shows how many times each level of recomputation occurred during the mouse dragging. It shows that the Level 1 recomputation accounts for a large share of the total computation while the Level 3 recomputation occurs only occasionally. Figure 4.2 shows the cost required for each level of recomputation. It is measured on the same FEM problem as the vibration analysis example in Section 4.3, tested with a 2.5-GHz CPU and 2.0 GB of RAM.

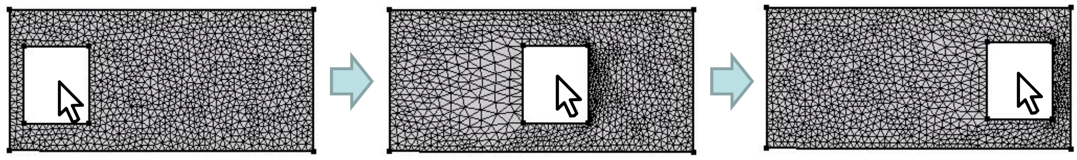


Figure 4.1: *An example modeling sequence used for the performance measurement. The user continuously drags the hole from left to right. The mesh consists of 1952 elements.*

Table 4.2: *The frequency of each recomputation during the example modeling sequence shown in Figure 4.1.*

	Level 1 (Relocation)	Level 2 (Reconnecting)	Level 3 (Reconstruction)
# of occurrences	117	39	3

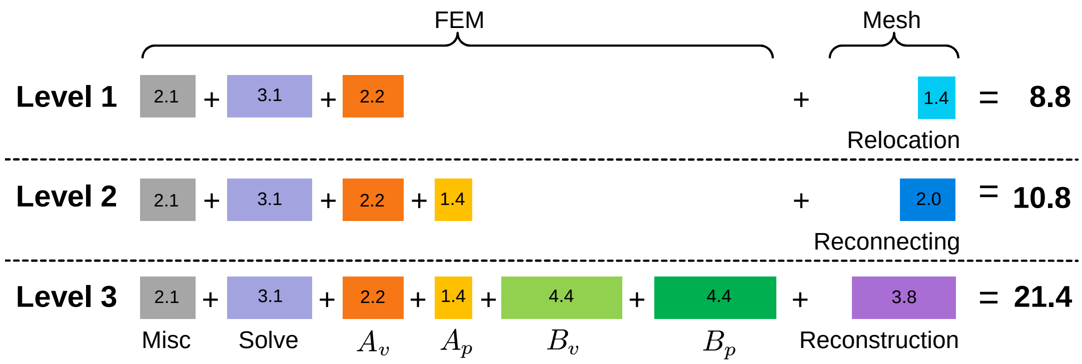


Figure 4.2: *The cost required for each level of recomputation (ms). The Level 3 recomputation is more than twice as expensive as the Level 1 recomputation, which is mainly due to the cost required for the construction of the preconditioner (B_v and B_p).*

4.3 Application Examples

We show several preliminary examples of applying a responsive FEM framework to typical 2D design problems. In each of these examples, the user interactively manipulates the shape of an object within a certain physical constraint and the system returns the analysis result in real time. We envision that these responsive simulations for geometric modeling will be useful for both early exploration of new design problems and refinement of designs that are almost finished. We present the current implementations primarily as a proof of concept to clarify this vision. They may not necessarily be useful for practical real-world applications; building practical applications based on these examples remains a subject for future work. Still, we believe that the current implementation is already useful for some end-user design problems as shown in the next section, and to teach the general principles of physical phenomena.

Vibration analysis of a structural object. In this example, a structural object is fixed to the ground that is shaking constantly at a certain frequency, causing the entire structure to deform (Figure 4.3). Resonance behavior appears when the user manipulates the object into a specific shape, one that would only be predictable through the use of simulation. We expect this application to be much more sophisticated in the future to help in the design of a building that would avoid collapsing due to resonance caused by an earthquake or wind.

Equation. The analysis is based on a nonstationary 2D linear solid without gravity:

$$\rho \ddot{\boldsymbol{u}} = \nabla \cdot \boldsymbol{\sigma}, \tag{4.1}$$

$$\boldsymbol{\sigma} = \lambda \left(\mathrm{tr} \varepsilon \right) \boldsymbol{I} + 2\mu \boldsymbol{\varepsilon}, \tag{4.2}$$

where $\boldsymbol{u}$ is the displacement, ρ is the density, $\boldsymbol{\sigma}$ is the Cauchy stress tensor, $\boldsymbol{\varepsilon}$ is the linearlized strain tensor, and λ and μ are the elastic Lamé coefficients. The time integration is based on the Newmark-β method.

Cantilever deformation. In this example, the leftmost part of a horizontal cantilever is fixed to a vertical wall while the remainder is free. The gravity causes the whole cantilever to deform (Figure 4.4). This application can possibly be of benefit in the design of an airfoil, in which the designer is most interested in the hydrodynamic performance of the shape after the deformation caused by gravity and wind pressure, rather than the original undeformed shape. Automatic optimization is usually used for this kind of problem when the goal shape is clearly defined. However, the user may often have only vague ideas about the goal shape and wishes to try various designs before deciding on one; continuous feedback can be very useful in such cases. Also note that the design shape can be used as an initial guess for the automatic optimization problem.

Equation. We solve the St.Venant–Kirchhoff material equation:

$$\boldsymbol{S} = \lambda \left(\mathrm{tr} \boldsymbol{E} \right) \boldsymbol{I} + 2\mu \boldsymbol{E}, \tag{4.3}$$

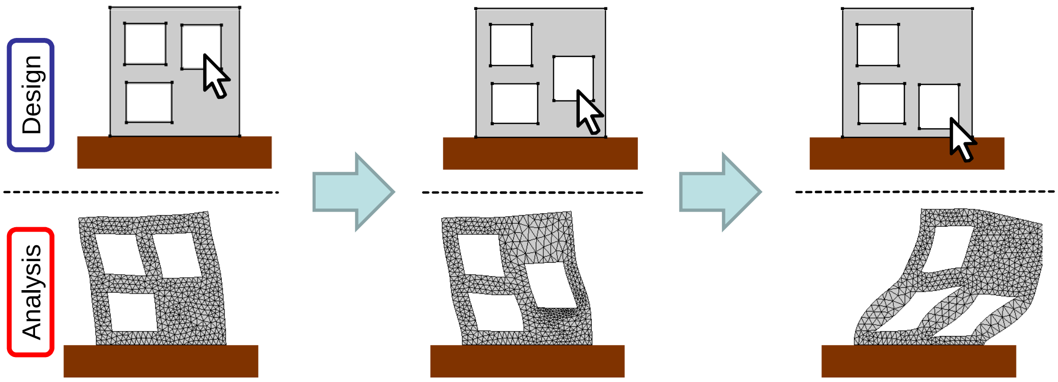


Figure 4.3: *Vibration analysis example. A structural object deforms due to the shaking movement of the ground. Notice that resonance occurs when the user moves the top right window toward the bottom, leading to a large destructive deformation. The displayed deformation is exaggerated for the purpose of visualization.*

where S is the second Piola–Kirchhoff stress tensor, E is the Green–Lagrange strain tensor. Both λ and μ are the same as in Equation 4.2.

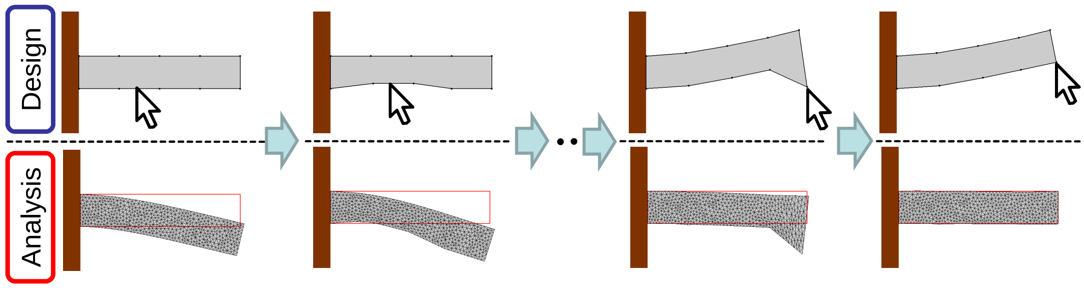


Figure 4.4: *Cantilever deformation. The user tries to fit the cantilever shape after deformation caused by gravity (bottom row) to a certain goal shape (shown in red lines) by continuously manipulating the undeformed shape (top row).*

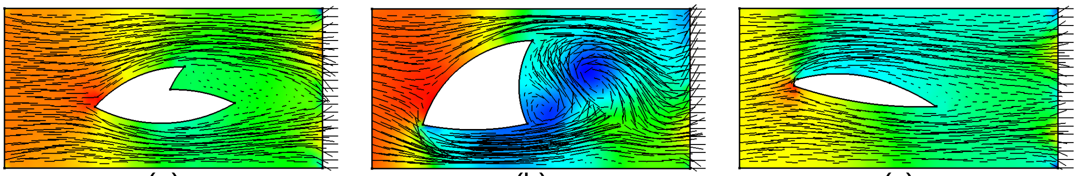
Fluid around an object. In this example, an object is placed inside a space filled with air, and a certain velocity of wind blows constantly from left to right, creating complex flow around the object (Figure 4.5). Depending on the object shape manipulated by the user, we can observe various kinds of phenomena such as boundary layer separation (Figure 4.5a), which can cause a stall, or a Karman vortex street (Figure 4.5b), which leads to an oscillation that may destroy the object. This application shows its potential utility for the design of various objects that are constantly exposed to a strong flow; this includes objects such as airfoils, car bodies, door mirrors, and air ducts.

Equation. We solve incompressible Newtonian flow stabilized with the SUPG-PSPG algorithm [156]:

$$\rho \frac{D\boldsymbol{v}}{Dt} = -\nabla p + \mu \nabla^2 \boldsymbol{v}, \tag{4.4}$$

$$\nabla \cdot \boldsymbol{v} = 0, \tag{4.5}$$

where $\boldsymbol{v}$ is the velocity, ρ is the density, p is the pressure, and μ is the viscous modulus. We used the implicit method for time integration.



(a) (b) (c)

Figure 4.5: *Fluid around an object. The velocity field is displayed as line segments, while the pressure is visualized as color contours. As the user manipulates the object shape, various phenomena can be observed such as boundary layer separation (a) and a Karman vortex street (b).*

Thermal fluid inside an object. In this example, some kind of fluid (e.g., water) fills an object (e.g., a teapot) whose bottom is heated while other boundaries are constantly cooled. We observe how the complex nonstationary behavior of the thermal fluid changes according to the object shape manipulated by the user (Figure 4.6). In addition to the design of a heat-efficient teapot, we expect this application could be useful for various problems concerned with thermal fluid phenomena such as the layout of room air conditioners or the design of a computer case.

Equation. We solve the Navier–Stokes equation with buoyancy proportional to the temperature, which is computed via a convection–diffusion equation:

$$\rho \frac{D\boldsymbol{v}}{Dt} = -\nabla p + \mu \nabla^2 \boldsymbol{v} + \rho \boldsymbol{g} \beta \left(T - T_0\right), \tag{4.6}$$

$$\nabla \cdot \boldsymbol{v} = 0, \tag{4.7}$$

$$\frac{DT}{Dt} = \alpha \nabla^2 T, \tag{4.8}$$

where T is the temperature, T_0 is the reference temperature, α is the thermal diffusivity, β is the volumetric thermal expansion ratio, $\boldsymbol{g}$ is the acceleration of gravity, and $\boldsymbol{v}$, ρ, p, and μ are defined as in Equation 4.4. We used the implicit method for time integration.

Performance. Table 4.3 summarizes the performance of our application examples. Because the frames per second (FPS) depends on the user manipulation (i.e., the FPS decreases when the user makes a large shape modification very quickly), we averaged the FPS measured during the user manipulation which is similar to that in Section 4.2.4 at a moderate speed. These results show that our data reuse scheme allows our application examples to run at a quite high frame rate that is sufficient for the

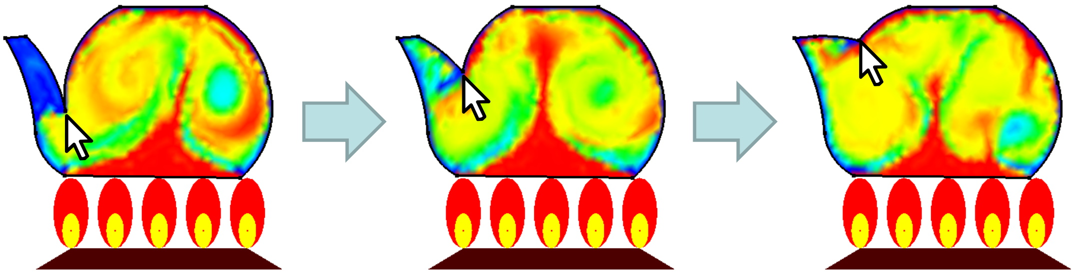


Figure 4.6: *Thermal fluid inside an object. The temperature is visualized as a color contour; blue and red correspond to low and high temperatures, respectively.*

interactive modeling. We also measured the FPS without data reuse (the last column). It shows that our method significantly improves the performance and is particularly effective for linear or stationary problems.

Table 4.3: *Performance of our application examples tested with a 2.5-GHz CPU and 2.0GB of RAM. The third and fourth columns show FPS with and without data reuse, respectively.*

Title	linear	stationary	#elem	FPS (reuse)	FPS (no reuse)
Vibration	yes	no	1962	105.0	41.8
Cantilever	no	yes	990	37.6	7.1
Fluid	no	no	1971	30.0	18.2
Thermal fluid	no	no	1938	21.3	14.3

4.4 Evaluation 1: Metallophone Design

In this section, we propose metallophone design system using real-time FEM eigenanalysis achieved by reuse algorithm.

4.4.1 Introduction

Each acoustic musical instrument has its own typical shape and appearance. Although the exterior may show subtle differences, the fundamental shape cannot be very different (e.g., metallophone plates are always rectangular). Although these shapes have become sophisticated through years of refinement, the appearance of musical instruments can be repetitive and characterless. Conveying character, or a message, through appearance is a major challenge for the makers of acoustic musical instruments. Because the shapes of acoustic musical instruments and their tones are inseparable, making designing them much more complex, acoustic musical instruments cannot have individual designs like electric guitars.

Figure 4.7: *A custom designed metallophone in the shape of a fish.*

We specially focus on metallophone design as a specific evaluation target of our Responsive FEM framework. We define a metallophone as a musical instrument that produces sounds via the vibration of metal plates struck by a mallet. Metallophone plates are usually rectangular because this practical shape makes it possible to analytically predict the instrument's tone [58]. Designing a metallophone with an arbitrary shape is difficult because of the complex relationship between shape and tone.

We describe a system for designing original metallophone with computational assistance. We utilize Responsive FEM framework that continuously update the simulation results in response to the designer's shape modifications. For the tone prediction, we applied FEM based eigen analysis. Real-time tone prediction feedback during editing provides the guiding principles for creating a better design while keeping the desired metallophone tone. We believe that our Responsive FEM framework is well suited to designing a metallophone of desired artistic shape and tone because this kind of highly constrained modeling naturally requires a tight integration of design and analysis.

4.4.2 User Interface

This section describes our metallophone design system from the user's point of view. The system basically operates as a two-dimensional (2D) modeling program in which the user interactively edits the shape of a metallophone plate using direct manipulation (Figure 4.9). FEM eigenfrequency analysis runs concurrently with the user manipulation and presents the predicted tone to the user with visual and audio feedback. The detail algorithm of FEM eigenfrequency analysis we used is described in Appendix A. The system also provides 3D graphics of the geometric deformation of the plate during vibration. In this way, the user, guided by artistic inspiration, can explore various plate shapes while continuously verifying that the desired tone is produced.

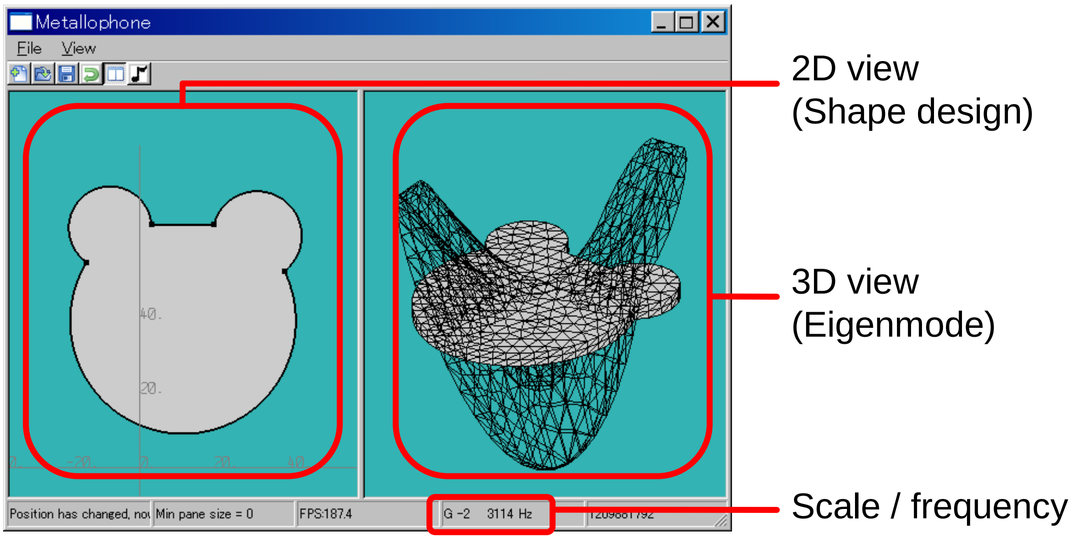


Figure 4.8: *Metallophone design system. The left window is used for the original 2D design, whereas the right window shows the analyzed eigen-mode as a 3D graphic. The tone is updated in real time with both audio and visual feedback for the user.*

Figure 4.9 shows a set of modeling operations provided by our system. A metallophone plate is represented as a closed area surrounded by straight lines and arcs, connected by corner vertices. The user drags the corner vertices to move them, and drags the arc to change its radius. The user can also add and delete corner vertices. Our current implementation does not support plates with holes. More complex curves, such as Bézier curves or NURBS, permit the creation of more diverse shapes; this is left to future development.

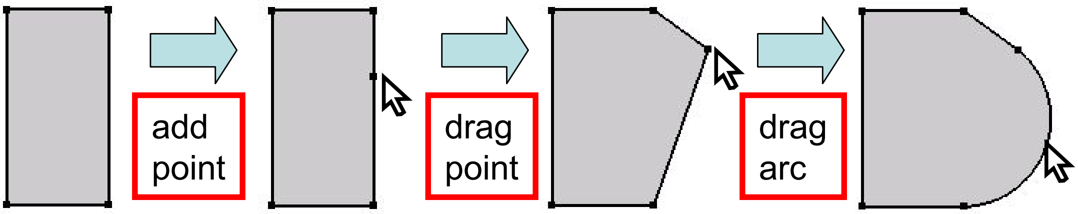


Figure 4.9: *The modeling operations of our system: adding and deleting a point, dragging an arc and a point.*

The system continuously predicts the tone that the plate will produce when struck with a mallet and presents it to the user during the editing process. Visual feedback is given as a numerical value (Figure 4.8, bottom) and audio feedback is provided in the form of a sine wave, with the speaker emitting the predicted tone. As the user drags the corner vertex, the tone from the speaker and the numerical value on the screen gradually change. Audio feedback allows the user to continuously monitor the tone while editing, and visual feedback is useful for verifying the exact value. Low tones tend to be produced by large plates (and vice versa), and the user can observe and learn this phenomenon during interactive editing, which facilitates the overall design process.

The system also predicts the geometric deformation of the plate during vibration and visually presents it to the user as a 3D graphic (Figure 4.8, right). The actual deformation is too small to be visible, so the system exaggerates it in the visualization. This helps the user decide where the plate should be attached to the base, because the plate should be attached at a point of minimal deformation. It is also useful for visually evaluating the quality of the vibrations. Vibrations perpendicular to the plate are desiable because they will be excited when struck by a mallet, whereas in plane vibrations are undesiable because they are unusual modes of oscilation (Figure 4.10). Thus, the user should strive for a shape that produces perpendicular vibrations.

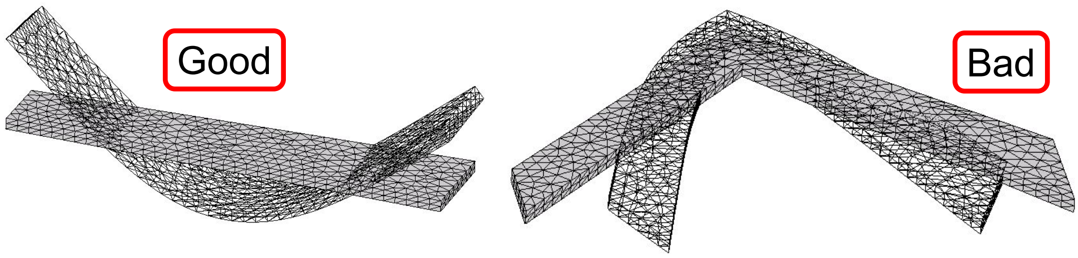


Figure 4.10: *Perpendicular vibration mode is desiable (left), whereas in plane vibration mode is undesiable (right).*

The system only supports the design of individual plate geometry, and physical construction must be done manually (Figure 4.11). The user may export the plate geometry to a DXF file and then send it to a wire-electrical discharge machine for the actual cutting of the metal plate. The metal plate can also be cut manually with a band saw. Some errors always occur during this process, and the finished metal plate will not produce exactly the same tone as the simulated plate. It is therefore necessary to adjust the tone by rasping the plate. Our system is also useful in this regard, because it is able to predict how the tone will change when particular edges are rasped. Finally, the user attaches the plate to a wooden board at the point suggested by the vibration shape analysis.

4.4.3 Results

System Performance. Table 4.4 summarizes the performance of our system. Because the performance depends on user operations, we measured the average number of frames per second during the continuous editing of shapes. In our current implementation, the number of elements is proportional to the area of the plate, because

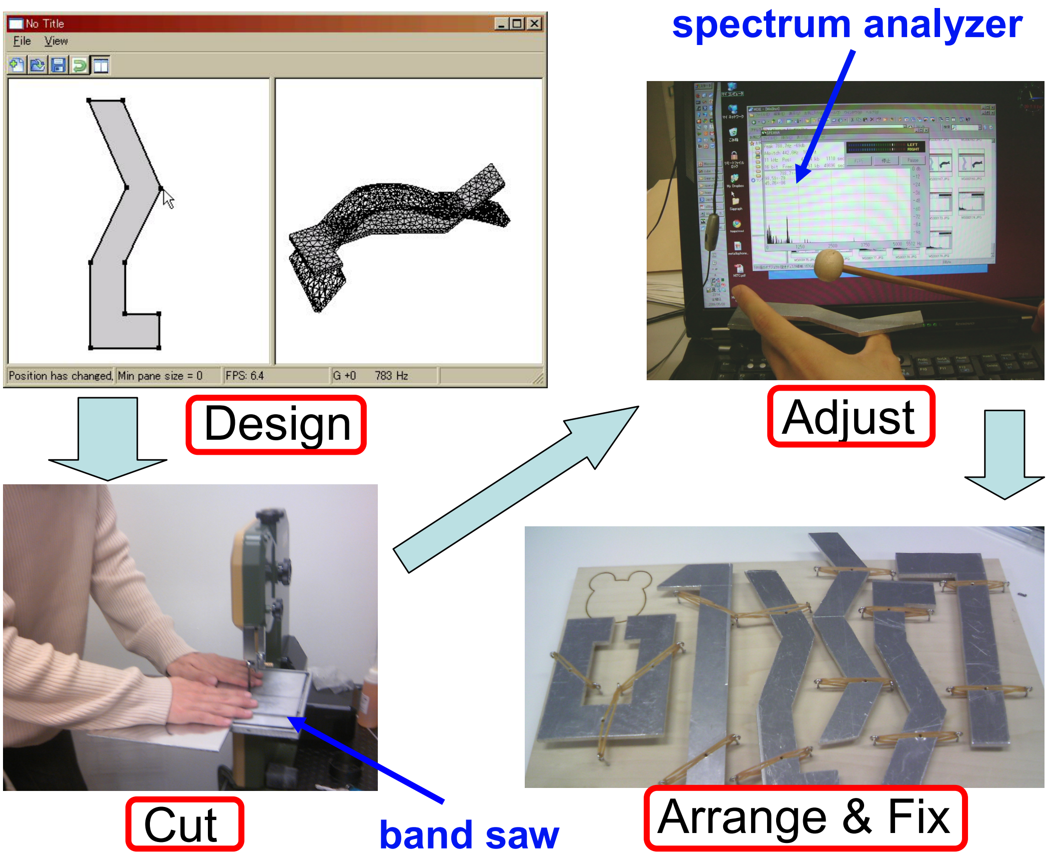


Figure 4.11: *The process of producing a metallophone: first, cutting the metal plate; second, adjusting the tone by rasping; finally, attaching the plate to a wooden board.*

we have limited the edge length to avoid element distortions. This causes a slight slowdown when the shape is large, but we consider it acceptable.

Table 4.4: *The performance of our system during interactive mnipulation, tested with a 2.5GHz CPU and 2.0GB RAM. The first column shows the size of the rectangular metallophone plates, the second shows the number of tetrahedral elements, the third shows the frame per second, the fourth shows the frequencies (Hz).*

Size (mm)	#Tetrahedra	FPS	Frequency (Hz)
100×30	2196	10.7	1931
150×30	3192	4.2	860
200×30	4524	3.3	494

User Experience. We asked a professional artist to design a metallophone using our system. The artist was allowed to work on the task freely and without time limitations and was provided instructions when required. We then used an electric discharge machine to manufacture an actual metallophone based on the artist's design. Figure 4.12 shows the shapes designed to correspond to the musical scale notes from C (523

Hz) to B (987 Hz). After the design of each pieces, we trimed and assembled the metallophone (Figure 4.7)

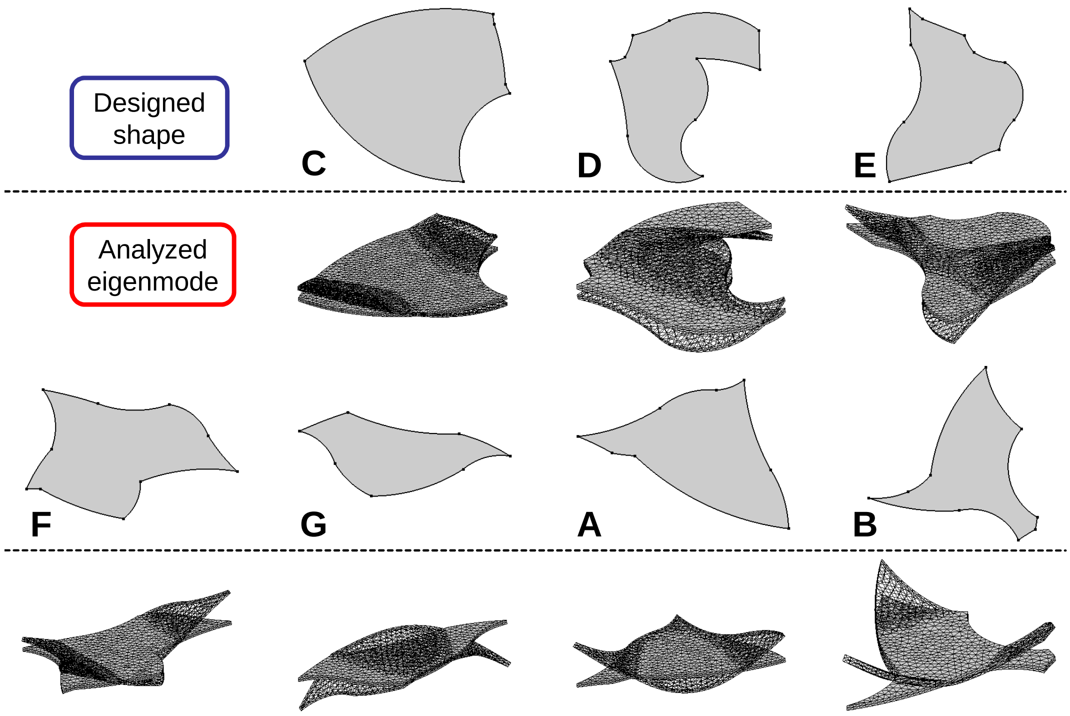


Figure 4.12: *Metallophone shapes designed by an artist. The upper row shows the designed 2D shapes, whereas the lower row shows their analyzed eigenmodes in 3D.*

The top three rows in Table 4.5 show that the frequencies of most of the pieces conformed well to one another for the target, the simulation, and the actual metallophone. To further improve the quality, we manually adjusted the tones of the actual metallophone pieces by trimming their edges (except for the piece corresponding to F). As mentioned in Section 4.4.2, our metallophone design system was also useful for this adjustment process, because it was able to predict the tone changes caused by edge trimming. The bottom row in Table 4.5 lists the frequencies of the actual metallophone after manual adjustment. Although inharmonic overtones make the timbre unclear, we found that the sounds produced by an actual metallophone were of acceptable quality for a hobbyist.

Table 4.5: *Target, simulated, measured, and adjusted frequencies (Hz) of the metallophone for each note in the scale, illustrating the accuracy of our analysis.*

Scale	C	D	E	F	G	A	B
Targeted	523	587	659	698	783	880	987
Simulated	525	588	661	699	786	880	989
Measured	506	604	621	698	787	860	993
Adjusted	523	587	659	698	783	881	987

We interviewed the artist afterward to obtain subjective feedback. The artist reported that the design took roughly 5 hours to complete, with most of the time being devoted to the C and D pieces. This was mainly because these lower tones required larger areas than the others, which greatly slowed the response of the analysis. One of the difficulties the artist encountered during the design process was needing to maintain an overall balance in shape among the pieces while keeping their tones true to the intended tone. This was a major design constraint. Another difficulty was that sometimes a small modification of the shape resulted in a large change tone, necessitating high responsiveness of the analysis. Finally, the artist noted that the relationship between the shape and tone of the metallophone was still difficult to understand, even after participating in this study. Nevertheless, the artist at least learned that larger pieces tend to produce lower tones, having previously believed the opposite to be true.

4.5 Evaluation 2: Bridge Design

The next study concerned the design of a bridge, with the aim of showing that responsive FEM can provide better support than traditional nonresponsive FEM for nonprofessionals in the design of objects with physically desirable properties.

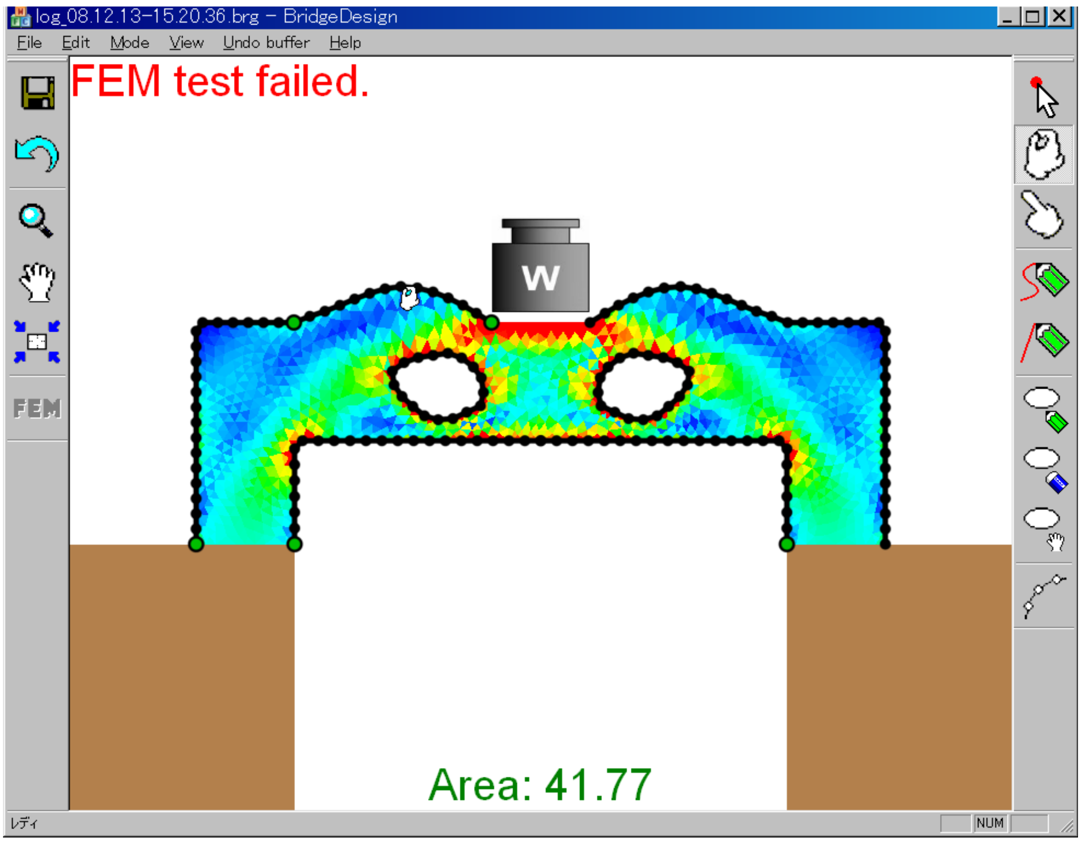


Figure 4.13: *A screenshot of our bridge design system.*

Task. The task given to the users was to design the 2D shape of a bridge to span a certain gap and support a certain weight on its center, as shown in the inset. Its strength was tested through FEM analysis with the physical model based on equivalent stress. The system displays the amount of stress being applied to each region as color contours (blue and red correspond to low and high stress, respectively) and judges whether the bridge passes the test. The users were asked to design a bridge that passes this strength test with as small an area as possible. In other words, the goal was to design a strong bridge with the least amount of material.

The shape design software used in this study provides a set of tools such as pushpin-and-pull curve editing [80], curve smoothing, and holes creation. The area of the bridge is always displayed during the design. The software has FEM analysis functionality with two modes: responsive FEM mode and nonresponsive FEM mode. In responsive FEM mode, the analysis is always performed during the user interaction (i.e., mouse dragging) and the result is updated in real time. In nonresponsive FEM mode, however, the analysis is performed only when the user completes the design and presses a button on a toolbar. The analysis result immediately disappears when the user changes the design. This mode simulates the way most existing FEM systems are used, in which the design process and the analysis process are completely separate, and switching between these two involves a great deal of tedious work such as file export/import and FEM parameter setup.

Experimental setup. Six university students majoring in art and design participated in the study, all of whom were unfamiliar with FEM techniques and material mechanics. These participants were split into two groups, A and B. Participants in group A used the responsive FEM mode, while participants in group B used the nonresponsive FEM mode. Experiments for these two groups were performed separately. Each group was first given a 15-min lesson on software usage, followed by a 30-min main design session. After that, participants in each group were asked to try the other FEM mode in a follow-up session, and their subjective feedback on the two FEM modes was collected.

Results. Figure 4.14 shows the smallest area of the bridge that passes the strength test for each participant in the main design session. We observed that participants who used responsive FEM generally achieved better results than those who used nonresponsive FEM. The most common subjective feedback was that responsive FEM is very useful when the user wants to make a small adjustment to see how it affects the analysis. Another interesting feedback was that the analysis result displayed during the design in responsive FEM mode could be too conspicuous, making a large design change difficult. Some participants even pointed out that shape design without responsive FEM may be more appropriate for initial design exploration.

We should emphasize that the nonresponsive FEM mode used in this study is already much more efficient than current commercial FEM tools, which require many time-consuming procedures such as switching between different tools and setting

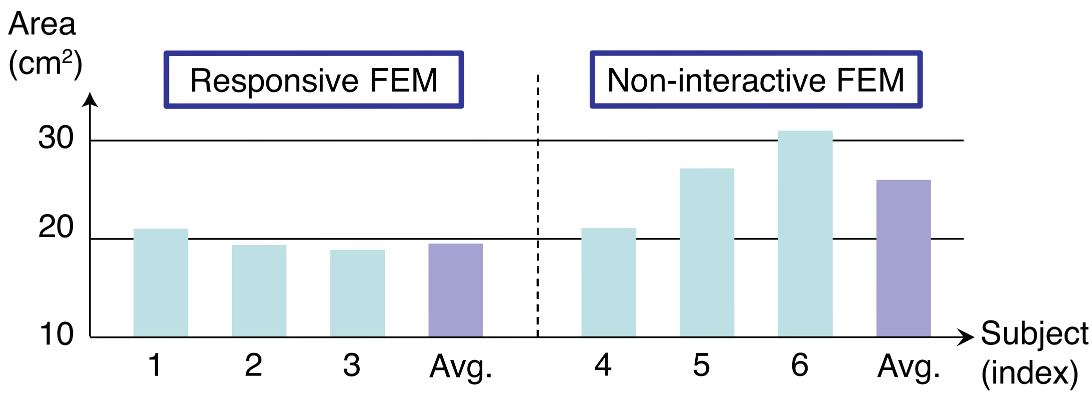


Figure 4.14: *Result of the bridge design study showing that subjects using responsive FEM generally achieved a better design.*

many parameters each time. This simple study obviously cannot *prove* anything but we believe that it at least *suggests* the potential of responsive FEM as a design aid for nonprofessionals, and it is our future work to perform more formal user studies.

4.6 Limitations and Future Work

Limitations. Most importantly, we need to extend our techniques to 3D so that they will be truly useful for many practical real-world problems. While solving linear systems in 3D itself is rather straightforward, the main challenge would be the continuous mesh update scheme in 3D. As noted by Labelle and Shewchuk [94], existing methods for improving tetrahedral mesh quality by continuously moving nodes and changing connectivity have yet to guarantee sufficient quality for accurate simulation.

Another limitation is that currently we can use first-order elements only. Higher-order elements are much more desirable for some problems such as bending of thin-walled structure. However, using them may be problematic in our approach because the reconnecting of the mesh would probably change the relationships between the nodes and prevent the matrix reuse. In addition, we have yet to try non-simplex elements (e.g., quadrilateral in 2D and hexahedron in 3D) that can be more appropriate than simplex elements (e.g., triangle in 2D and tetrahedron in 3D) depending on the problems, although the reconnecting of such non-simplex meshes without adding and deleting points is generally known to be difficult.

Our approach cannot be applied to history-dependent problems such as plasticity processing because the solution in these cases needs to be computed sequentially from the initial state, and our data reuse scheme is inappropriate for that purpose.

Future work. We plan to test reusing various kinds of data other than the preconditioner matrix. This could include node reordering, which will also improve the responsiveness of analysis.

One future direction is to make the system more actively guide the design process using the result of simulation. For example, it would be useful if the system could assist the user design shapes that satisfy certain constraints (e.g., certain stress limits in certain areas) by guiding the user manipulation with instructions and suggestions whenever the user makes a design change that will not satisfy these constraints. We explored this direction as described in Section 6.

Another direction would be to let the user interactively control the simulation accuracy. We use fixed criteria for the speed–accuracy trade-off, but the user may want more explicit control over it during the design (i.e., the user may want more accuracy than speed in the design refinement stage, and vice versa). We also assume the homogeneous mesh density, but it would be useful if the user could control the simulation accuracy locally by manipulating the local mesh density. This would help the user examine the analysis results more closely in the specific region of interest, which would be difficult with existing automatic error estimation methods.

Chapter 5

Acceleration of Static FEM Simulation Response

This chapter introduces acceleration techniques for the response of static FEM simulations to design change. There are two categories of FEM simulations: static simulations and dynamic simulations. Static FEM simulations are used to simulate stationary solutions to static scenarios. Many simulations, such as those of architectural deformation under constant gravity or converged heat distributions under constant boundary conditions, fall into this category. Dynamic FEM simulations are used to simulate transitional time-developing phenomena, such as turbulent flow, vibrations, and wave propagation. The acceleration techniques described in the previous chapter (Chapter 4) can be used for both static and dynamic FEM simulations with up to several thousand degrees of freedom. In this chapter, we describe further acceleration techniques that enable us to handle hundreds of thousands of degrees of freedom at an interactive rate.

A static FEM simulation solves for static equilibrium conditions (i.e., all forces are in balance). We leverage the observation that static equilibrium can be represented implicitly as a function of the rest shape and the deformed shape. We then investigate the linear tangent space of the implicit manifold to find a leaner relationship between the input rest shape and the output static simulation results. First-order approximation of simulation response is often called design sensitivity analysis. Typically, design sensitivity analysis is used for offline shape optimization, but we propose a procedure for employing it in an interactive context. We extend the technique from linear response to nonlinear response by utilizing progressively cached local approximations. We demonstrate our approach with a clothing pattern design system called *Sensitive Couture*. This chapter first presents the basic *first-order approximation* algorithm, and then describes the specific interface and algorithm required for the clothing pattern design system.

5.1 Introduction

Static deformation is important in the design of artificial objects. For example, in bridge design, the designer first considers static equilibrium deformation under constant gravity, and then investigates the dynamic deformation around equilibrium. Thus, the design is often captured in a static state. However, it takes time for static deformation to converge to equilibrium. To obtain the results of a static simulation, it is often necessary to run dynamic simulations or conduct a lengthy series of iterations using the Newton-Raphson method. Hence, the designer has to wait until the simulation provides static results from the input rest shape. The "Responsive FEM" framework accelerates the response to changes in the rest shape, but the response speed depends heavily on the complexity of the problem, making it difficult to apply the technique to large-scale static simulations.

In this chapter, we propose a novel *first-order approximation* algorithm to solve this problem. The static state in an FEM simulation is characterized by the condition that the residual (i.e., the remaining unbalanced forces) is equal to zero. In a space with a large number of dimensions, where the rest shape and the deformed shape have separate elements, the equilibrium state can be represented as an implicit surface on which the residual is zero. We consider a tangent plane to this iso-surface to determine a linear relationship between the shape and the static solution. This technique is called design sensitivity analysis. Using this approximate representation of static shape, the system can provide the user with a quick response, even when the scale of the simulation is relatively large. At the beginning of the user's direct manipulation, the system computes the first-order approximation for the initial design. As the user continues to work, the system progressively updates the approximation by sampling from the tangent planes at several points on the surface representing equilibrium.

Clothing pattern design. We focus on clothing pattern design as a target application. An article of clothing is created in accordance with a drafted pattern, which serves as a blueprint for the final product. The pattern specifies how to cut the 2D fabrics and stitch them together so that the resulting article of clothing fits a target 3D human body. However, the design of clothing patterns is not intuitive, since the relationship between a 2D clothing pattern and the stitched 3D clothing drape is not obvious. The drape of a garment over a curved body is affected by frictional contact, and mapping from a 2D representation to a 3D representation is complex and nonlinear. Thus, the clothing design process requires tiresome repetitions of drafting, verification by actual sewing, and revision. Even experienced dressmaking teams go through many repetitions, in which the designer conceptualizes 3D forms in sketches and the pattern-maker drafts precise 2D outlines (see Figure 5.1)

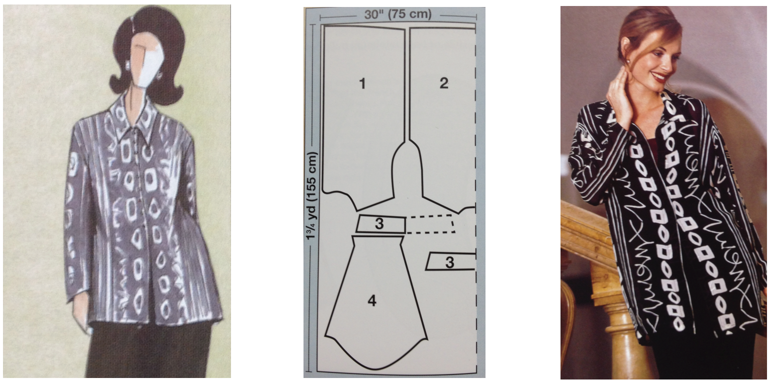


Figure 5.1: *An article of clothing is first sketched, then a pattern is designed by a pattern-maker, and finally we can obtain wearable clothing by cutting and sewing according to the pattern [52].*

Clothing design and computer graphics. Although both the 2D ("intrinsic") and 3D ("extrinsic") perspectives are innate aspects of garment design, existing software tools are limited to one view or the other. The situation is obvious in the computer animation industry. In general, artists either *sculpt* in 3D or *draft* in 2D (but not both), and the choice is typically determined by the software used at the studio. Sculpting tools provide a familiar interaction for a computer graphics artist, but do not account for the special (low Gaussian curvature) structure of draped fabric, and thus the results appear less natural. Sometimes the 3D mesh is computationally flattened [49], a process that alters the 3D form and consumes time. While drafting tools ensure that the final form is natural, the artist must learn to draft 2D patterns, and must await the results of draping computations that produce the 3D form [167]. In summary, existing flattening (3D→2D) and draping (2D→3D) computations are similar in their effect to the time and material impediments of manual dressmaking: they divide the design process into discrete tasks and inhibit a free-flowing creative process.

***Sensitive Couture* (SC) for interactive clothing pattern design** We present a novel approach to garment design that leverages responsive static FEM simulations. Our software tool provides a continuous, interactive, natural design modality. The 2D design and the 3D draped form receive equal status, are simultaneously visible, and seamlessly correspond to one another. The artist may interactively edit the 2D design and immediately observe how these changes affect the 3D form. There are a number of technical obstacles to incorporating online cloth simulation into the design process. Existing high-resolution simulation codes are not yet fast enough to maintain an interactive correspondence between the 2D and 3D views. The simulation must account for geometric nonlinearity and frictional contact while remaining stable, even during rapid user input. To address these challenges, our design tool employs a combination of techniques, including (i) fast prediction of 3D forms from cached shapes, using sensitivity analysis and generalized moving least squares, (ii) fast invisible remeshing, using positive-mean-value coordinates to accommodate arbitrary revisions of the pattern boundary, and (iii) stable and accurate cloth modeling, using an isometric bending model, a modified St. Venant-Kirchhoff membrane element, and progressive refinement. The novel use of sensitivity analysis to enable interactive-rate synchronization of the 2D and 3D perspectives is reflected in the tool's name, *Sensitive Couture* (SC).

Our system accelerates the response of a static FEM simulation by using design sensitivity analysis based on first-order estimation of simulation results with respect to the design parameters.

The response of a static simulation usually exhibits strong nonlinearity relative to the design. At the same time, naïve design sensitivity analysis predicts only linear simulation responses. Thus, predictions obtained via design sensitivity analysis may not provide accurate approximations for nonlinear simulation responses. We overcome this problem by caching many solutions and sensitivities online, and synthesizing a desired solution by interpolating them. In this section, we first explain the standard static FEM simulation technique (Section 5.1.1) and design sensitivity analysis (Section 5.1.2). Then we describe the computation of approximate static simulation results using sensitivity analysis (Section 5.1.3). A two-dimensional mesh-manipulation technique is described in Section 5.1.4. Finally, we illustrate nonlinear augmentation based on interpolation of many cached solutions and sensitivities (Section 5.1.5).

5.1.1 Standard Static FEM Simulation Methods

In this subsection, we describe the standard technique for computing a static FEM solution. A user-designed shape is given as a collection of points in r-dimensional space, and the FEM simulation computes a solution in the form of a spatial function that yields an s-dimensional value corresponding to each point. Note that r and s are not always equal. If we compute a scalar solution for a three-dimensional shape, $r = 3$ and $s = 1$. If we compute the drape of a two-dimensional flat sheet that deforms in three-dimensional space, $r = 2$ and $s = 3$.

In an FEM simulation, a shape is discretely represented by a mesh, which consists of a set of vertices and their connectivity. Here, we denote the number of vertices in the mesh by n. The positions of all the nodes in the mesh can be represented as an rn-dimensional vector concatenating all the coordinates of the node positions, denoted by $\mathbf{X} \in \mathbb{R}^{rn}$. The vector $\mathbf{X}$ is called the *design vector*. We can change the design to a certain extent by changing the initial vertex positions $\mathbf{X}$ without changing the connectivity of the mesh. We also represent the spatial solution function as a long vector concatenating all the values of the solution at the n mesh vertices, denoted by $\mathbf{x} \in \mathbb{R}^{sn}$. This vector is called a *solution vector*. In describing the shape and solution with $\mathbf{X}$ and $\mathbf{x}$, respectively, we follow the standard notation employed in

FEM simulations of nonlinear continuum mechanics[29, 18].

Static FEM is a type of variational procedure, and thus it provides a solution that minimizes a given functional W. When FEM is used to solve for the deformation of an elastic body, this functional W encodes the total energy of the body, and the procedure yields a solution that minimizes the total energy. In the FEM discretization, the solution can be represented in the discretized form $\mathbf{x}_{solution} \in \mathbb{R}^{sn}$, just as the vector $\mathbf{x}$ represents the function. FEM finds a solution that minimizes the functional $W(\mathbf{X}, \mathbf{x}) : \mathbb{R}^{rn+sn} \rightarrow \mathbb{R}$ over all the possible values of the discretized function, as follows:

$$\mathbf{x}_{solution} = \arg\min_{\mathbf{x} \in \mathbb{R}^{sn}} W(\mathbf{X}, \mathbf{x}). \tag{5.1}$$

We assume that the functional W is smooth with respect to the design vector $\mathbf{X}$ and the solution vector $\mathbf{x}$, and that the solution is unique. With these assumptions, solving Equation 5.1 is equivalent to finding $\mathbf{x}_{solution}$ such that the functional W takes its extremal value. When the functional W takes its extremal value, its gradient with respect to the vector $\mathbf{x}$ (with any number of degrees of freedom) is zero. The gradient of W with respect to the vector $\mathbf{x}$ is usually called the residual vector, and is denoted by $\mathbf{R} : \mathbb{R}^{rn+sn} \rightarrow \mathbb{R}^{sn}$. Hence the solution $\mathbf{x}_{solution}$ is obtained such that the residual vector $\mathbf{R}$ is zero:

$$\mathbf{R}(\mathbf{X}, \mathbf{x}_{solution}) = \left. \frac{\partial W(\mathbf{X}, \mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_{solution}} = 0. \tag{5.2}$$

Note that the solution vector $\mathbf{x}_{solution}$ and the design vector $\mathbf{X}$ are coupled to make the residual zero, and thus the solution vector $\mathbf{x}$ changes in accordance with the position vector $\mathbf{X}$ of the initial nodes. In a typical FEM simulation, the residual vector consists of two components:

$$\mathbf{R}(\mathbf{X}, \mathbf{x}) = \mathbf{F}(\mathbf{X}, \mathbf{x}) - \mathbf{Q}(\mathbf{X}, \mathbf{x}), \tag{5.3}$$

where $\mathbf{F}$ is an external force vector. This term arises from a gravitational force or from boundary conditions such as external forces. The term $\mathbf{Q} \in \mathbb{R}^{rn}$ denotes an internal force vector that arises from internal forces such as elastic forces. In an elastic simulation, a zero residual vector $\mathbf{R}$ indicates that all forces on the nodes are in balance, and thus the system is in equilibrium.

Static FEM can be thought of as a relationship between the three vectors $\mathbf{x}$, $\mathbf{X}$, and $\mathbf{R}(\mathbf{X}, \mathbf{x})$. Figure 5.2 illustrates the simple geometrical situation in which the vectors $\mathbf{x}$, $\mathbf{X}$, and $\mathbf{R}$ are all one-dimensional. Each combination of $\mathbf{x}$ and $\mathbf{X}$ satisfying $\mathbf{R}(\mathbf{X}, \mathbf{x}) = 0$ is a static solution. Thus, each static FEM solution is in the zero-level set $\mathbf{R}(\mathbf{X}, \mathbf{x}) = 0$ in the product space of $\mathbf{X}$ and $\mathbf{x}$, which provides an implicit representation of the solution.

In a typical FEM static simulation, we fix the design vector $\mathbf{X}$ and find a solution $\mathbf{x}$ that makes the residual $\mathbf{R}$ equal to zero. This is usually accomplished via the Newton-Raphson iteration method. We find $\mathbf{x}_{solution}$ in Equation 5.2 by starting from an initial

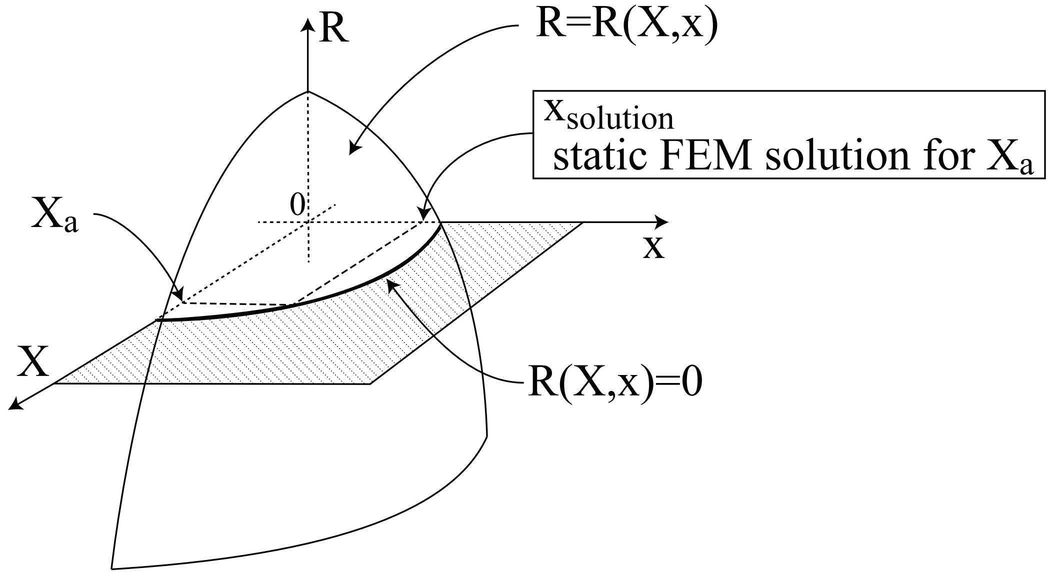


Figure 5.2: *Implicit interpretation of static FEM. The combination of a static FEM solution* $\mathbf{x}$ *and its design* $\mathbf{X}$ *lies in the zero-level set* $\mathbf{R} = 0$.

value $\mathbf{x}_0$, and updating $\mathbf{x}$ iteratively to obtain $\mathbf{x}_1, \mathbf{x}_2, \ldots$ such that $\mathbf{x}_i$ converges to $\mathbf{x}_{solution}$. In the standard Newton-Raphson method, the algorithm updates the solution as follows:

$$\left.\frac{\partial \mathbf{R}(\mathbf{X}_a, \mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}_i} \Delta \mathbf{x}_i = -\mathbf{R}(\mathbf{X}_a, \mathbf{x}_i), \tag{5.4}$$

$$\mathbf{x}_{i+1} = \Delta \mathbf{x}_i + \mathbf{x}_i. \tag{5.5}$$

In each iteration of the Newton-Raphson method, we compute the first-order approximation of the residual, and update the value $\mathbf{x}$ so that the approximated residual converges to zero. This procedure takes time, because in each iteration we must compute the inverse of a square matrix of order sn. Figure 5.3 presents a geometric interpretation of a static FEM solution using the Newton-Raphson method. On the left side of Figure 5.3, we show the cross-section of the residual $\mathbf{R}(\mathbf{X}, \mathbf{x})$ in a plane $\mathbf{X} = \mathbf{X}_a$. The static solution can be found where the cross-sectional curve crosses the plane $\mathbf{R} = 0$. On the right side of Figure 5.3, we show how to use the Newton-Raphson method to find a static solution. For each iteration, we first compute the first-order approximation of the residual $\mathbf{R}$ at $\mathbf{x}$, which is equivalent to computing the tangent line to the residual $\mathbf{R}$ at $\mathbf{x}_i$. We then update $\mathbf{x}$ to the point where the tangent line crosses the plane $\mathbf{R} = 0$.

If the nonlinearity of $\mathbf{R}$ in a static simulation is very high, the naïve Newton-Raphson iterations do not converge quickly. The nonlinearity tends to be very high in a scenario such as a large deformation of a thin shell. This creates a problem because the Newton-Raphson method finds a solution by using local first-order approximations, resulting in a prediction that can be very far from the actual solution. If a prediction obtained from the Newton-Raphson method is far from the solution,

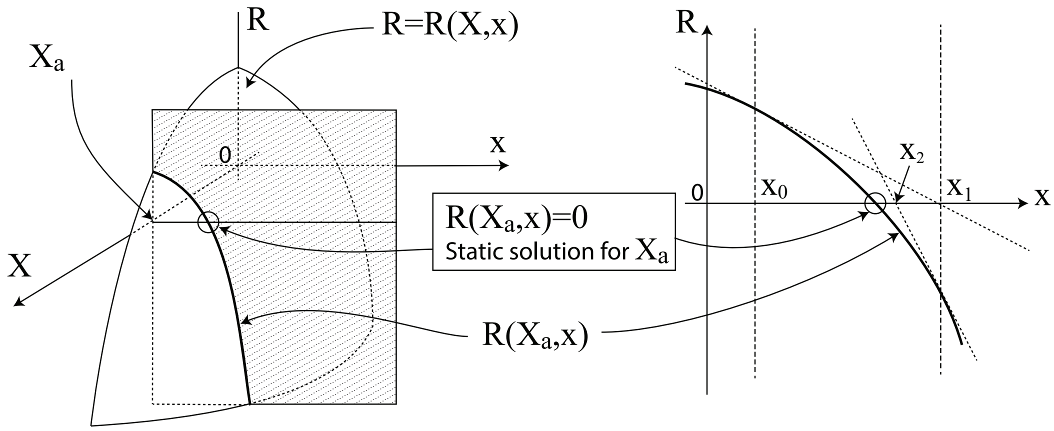


Figure 5.3: Left: *a cross-section (in the plane* $\mathbf{X} = \mathrm{x}_a$*) of the implicit surface* $\mathbf{R}$. Right: *the Newton-Raphson iterations yield a static FEM solution.*

more iterations are required and the convergence is slowed. To improve the convergence speed of static FEM, a variant of the Newton-Raphson method called *dynamic relaxation* [47] has been studied. The dynamic relaxation technique computes a static solution by solving an equilibrium state of a damped dynamic simulation. More details on dynamic relaxation can be found in [164, 35].

5.1.2 Design Sensitivity Analysis

In this subsection, we briefly describe the fundamentals of standard design sensitivity analysis. Design sensitivity analysis computes a first-order approximation of the relationship between the static FEM solution x and the design $\mathbf{X}$. Suppose we have a paired design vector $\mathbf{X}$ and solution vector x satisfying $\mathbf{R}(\mathbf{X}, \mathrm{x}) = 0$. When the design is slightly altered to $\mathbf{X} + \Delta\mathbf{X}$, the residual vector is no longer zero. We consider updating the solution vector from x to $\mathrm{x} + \Delta\mathrm{x}$ so that the change in the solution vector counterbalances the change in the design, and the residual is still zero:

$$\mathbf{R}(\mathbf{X} + \Delta\mathbf{X}, \mathrm{x} + \Delta\mathrm{x}) = 0. \tag{5.6}$$

We then expand Equation 5.6 in a Taylor series to obtain the first-order approximation:

$$\mathbf{R}(\mathbf{X}, \mathrm{x}) + \frac{\partial\mathbf{R}}{\partial\mathbf{X}}\Delta\mathbf{X} + \frac{\partial\mathbf{R}}{\partial\mathrm{x}}\Delta\mathrm{x} \simeq 0. \tag{5.7}$$

The first term of Equation 5.7 is zero, since we assume that the combination of $\mathbf{X}$ and x makes the residual vector zero.

When we consider an infinitesimal change in the design vector $\Delta\mathbf{X}$, the approximation of Equation 5.7 yields a linear relationship S between $\mathbf{X}$ and x:

$$\mathrm{S} = \frac{\partial\mathrm{x}}{\partial\mathbf{X}} = -\left(\frac{\partial\mathbf{R}}{\partial\mathrm{x}}\right)^{-1}\left(\frac{\partial\mathbf{R}}{\partial\mathbf{X}}\right), \tag{5.8}$$

where S is encoded by a $sn \times rn$ *design sensitivity matrix*. Note that the square matrix $\partial\mathbf{R}/\partial\mathbf{x}$ is the usual *stiffness matrix* associated with the nonlinear statics problem in Equation 5.4. If the statics problem has a nonsingular stiffness matrix, then $\partial\mathbf{R}/\partial\mathbf{x}$ has full rank. In design sensitivity analysis, we ensure that the stiffness matrix is always a regular matrix. The only difference between sensitivity analysis and a typical FEM simulation is the matrix $\partial\mathbf{R}/\partial\mathbf{X}$, which must be computed especially for the sensitivity analysis.

Sensitivity analysis is based on the *implicit function theorem* [74], which states that if $\mathbf{R}$ is continuously differentiable, $\mathbf{X}_a$ and $\mathbf{x}_{solution}$ satisfy $\mathbf{R}(\mathbf{X}_a, \mathbf{x}_{solution})$, and the Jacobian matrix $\partial\mathbf{R}/\partial\mathbf{x}$ is invertible, then there exists an open set $\mathbf{U}$ containing $\mathbf{X}_a$, an open set $\mathbf{V}$ containing $\mathbf{x}_{solution}$, and a unique continuously differentiable function $\hat{\mathbf{x}} : U \to V$ such that:

$$\{\mathbf{X}, \hat{\mathbf{x}}(\mathbf{X}) | \mathbf{X} \in U\} = \{(\mathbf{X}, \mathbf{x}) \in \mathbf{U} \times \mathbf{V} | \mathbf{R}(\mathbf{X}, \mathbf{x}) = 0\}. \tag{5.9}$$

Furthermore, this theorem guarantees that Equation 5.8 holds. The implicit function theorem enables us to construct a local one-to-one map between a design vector $\mathbf{X}$ and a solution vector $\mathbf{x}$, valid in the neighborhood of the solution, such that the first-order approximation of the map is given by Equation 5.8.

Figure 5.4 presents a geometric interpretation of design sensitivity analysis. The left side of Figure 5.4 shows a cross-section of the function $\mathbf{R}(\mathbf{X}, \mathbf{x})$ in the plane $\mathbf{R} = 0$. The cross-sectional curve provides the relationship between the design vector $\mathbf{X}$ and the solution vector $\mathbf{x}$. The right side of Figure 5.4 shows the tangent line to the cross-sectional curve that provides the first-order approximation of the relationship between the design vector $\mathbf{X}$ and the solution vector $\mathbf{x}$.

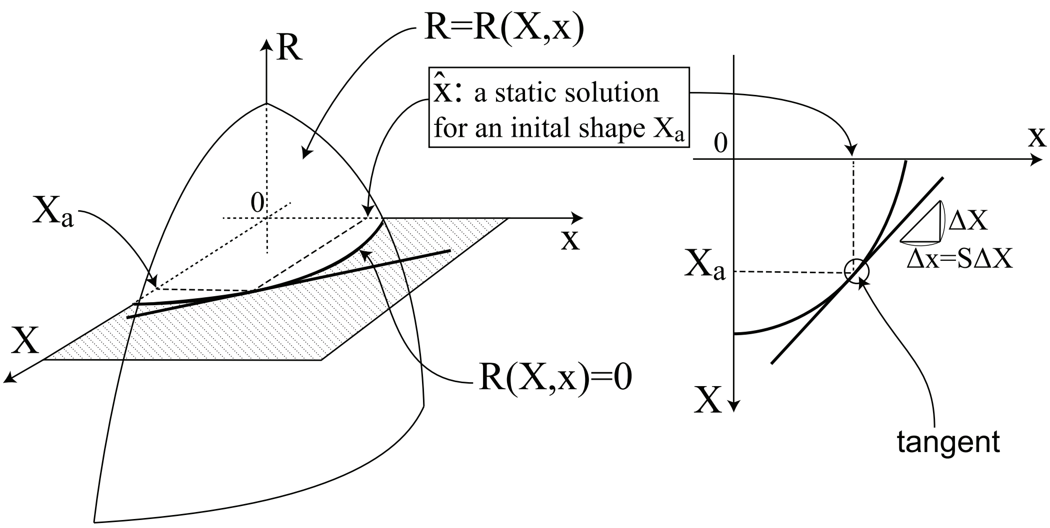


Figure 5.4: *A geometric interpretation of sensitivity analysis.*

Suppose the design vector $\mathbf{X}$ is parameterized with $\#param$ design parameters $\mathbf{q} = \{q_1, q_2, \ldots, q_{\#param}\}$. We can compute the sensitivity with respect to the design parameters $\mathbf{q}$ instead of the design vector $\mathbf{X}$ itself. The use of design parameters is convenient in design sensitivity analysis, because the dimensionality is significantly lower; the design vector has sn degrees of freedom, whereas the design parameters include only $\#param$ free parameters. In ordinary FEM simulations, sn is typically large (on the order of tens of thousands or hundreds of thousands), and it is not practical to compute Equation 5.8 directly. We assume that the design vector $\mathbf{X}$ is continuously parameterized with respect to these parameters $\mathbf{q}$, and thus differentiable with partial derivatives $\partial\mathbf{X}/\partial\mathbf{q}$. By applying the chain rule, we can obtain the design sensitivity with respect to each of the design parameters, as follows:

$$\mathbf{s}_i = \frac{\partial \mathbf{x}}{\partial q_i} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}\frac{\partial \mathbf{X}}{\partial q_i} = -\left(\frac{\partial \mathbf{R}}{\partial \mathbf{x}}\right)^{-1}\left(\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right)\frac{\partial \mathbf{X}}{\partial q_i} = -\left(\frac{\partial \mathbf{R}}{\partial \mathbf{x}}\right)^{-1}\mathbf{f}_i, \tag{5.10}$$

where $\mathbf{s}_i$ and $\mathbf{f}_i = (\partial\mathbf{R}/\partial\mathbf{X})\,(\partial\mathbf{X}/\partial q_i)$ are both vectors of dimension sn. We refer to the vector $\mathbf{s}_i$ as the *sensitivity mode*. Note that in Equation 5.10, we use Equation 5.8 for the transformation from the second term to the third term. In the computation of Equation 5.10, we first compute the vector $\mathbf{f}_i$. The vector $\mathbf{f}_i$ encodes the variation of the residual $\mathbf{R}$ vector with respect to the design parameter q_i. The matrix $\partial\mathbf{R}/\partial\mathbf{x}$ has dimension $sn \times rn$, and is too large to compute and store in memory. Instead, we evaluate $\mathbf{f}_i$ *element-wise*, in the same manner as the residual vector $\mathbf{R}$:

$$\mathbf{f}_i = \sum_e \frac{\partial \mathbf{R}_e}{\partial \mathbf{X}}\frac{\partial \mathbf{X}}{\partial q_i}, \tag{5.11}$$

where $\mathbf{R}_e$ is a residual vector element, whose size is s times the number of nodes belonging to the element. $\mathbf{f}_i$ is computed for each of the elements, and the results are then assembled for efficient computation. The linear system

$$-\left(\frac{\partial \mathbf{R}}{\partial \mathbf{x}}\right)\mathbf{s}_i = \mathbf{f}_i \tag{5.12}$$

is then solved via a linear solver to compute the sensitivity mode. Note that the coefficient matrix in Equation 5.12 is independent of i. Hence, once the coefficient matrix has been factored, we can reuse the factored matrix for all $i = 1, \ldots, \#param$ to solve the linear system. Factorization methods such as LU factorization take considerable time, but once the matrix has been factored, we can accurately and quickly solve the system. On the other hand, incomplete factorization methods such as incomplete LU factorization require relatively little computational time compared to complete factorization methods, and significantly accelerate the iterative solver in finding the sensitivity modes. Moreover, the coefficient matrix is exactly same as the matrix used in the Newton-Raphson iterations of the static FEM simulation in Equation 5.4. Thus, if the coefficient matrix in Equation 5.4 is factored completely or incompletely for static FEM simulation, we can reuse that factored matrix in the sensitivity mode computation. In the implementation of this equation, we can also reuse most of the coding resources to find the static FEM solution for the sensitivity modes.

Previously, we described a method of obtaining the design sensitivity (the first-order approximation of a change in the solution with respect to a change in the design) by solving Equation 5.8. This technique is called *analytical* design sensitivity analysis, because the design sensitivity is obtained exactly. However, there are two other methods of obtaining the design sensitivity numerically. One of these is the finite difference method (FDM), in which the system computes a static solution $\hat{\mathbf{x}}$ with a design vector that is slightly perturbed with respect to each of the design parameters, and then computes the gradient numerically as follows:

$$\mathbf{s}_i = \frac{\partial \mathbf{x}}{\partial q_i} \simeq \frac{\hat{\mathbf{x}}(\mathbf{X}(\mathbf{q} + \epsilon q_i \mathbf{e}_i)) - \hat{\mathbf{x}}(\mathbf{X}(\mathbf{q}))}{\epsilon}, \tag{5.13}$$

where $\epsilon \in \mathbb{R}$ is a small real number, $\mathbf{e}_i$ is a unit vector in which the ith component is all ones and the rest of the components are zeros, $\mathbf{q} + \epsilon q_i \mathbf{e}_i$ is a set of design parameters perturbed with respect to the ith design parameter, and $\hat{\mathbf{x}}(\mathbf{X}(\mathbf{q} + \epsilon q_i \mathbf{e}_i))$ is a static solution with respect to the perturbed design parameters. FDM allows a simple formulation, but there are many drawbacks. In FDM sensitivity analysis, the system must solve a nonlinear FEM simulation iteratively via the Newton-Raphson method, given by Equation 5.4. In a nonlinear FEM simulation, the coefficient matrix changes, and we must factor the matrix repeatedly to compute the static solution, resulting in a very high computational cost. Moreover the accuracy of the algorithm is heavily dependent on the choice of the parameter ϵ. A large value of ϵ tends to introduce approximation errors, while a small value of ϵ may introduce floating point errors. The second numerical method for obtaining the design sensitivity involves approximating the second term on the right-hand side of by FDM, as follows:

$$\mathbf{s}_i \simeq -\left(\frac{\partial \mathbf{R}}{\partial \mathbf{x}}\right)^{-1} \frac{\mathbf{f}_i(\mathbf{X}(\mathbf{q} + \epsilon q_i \mathbf{e}_i)) - \mathbf{f}_i(\mathbf{X}(\mathbf{q}))}{\epsilon}. \tag{5.14}$$

This technique is more promising than the first one. The cost of computing the sensitivity modes is comparable to that of the analytical method. However, the choice of ϵ still depends heavily on the target problem, and thus it is difficult to choose a desirable value of ϵ in most cases.

5.1.3 Fast Approximation Using Sensitivity Analysis

In this subsection, we explain how to use a mouse to interact with the design while receiving real-time feedback from an FEM simulation, leveraging the acceleration obtained from the sensitivity analysis (which we briefly described in Section 5.1.2). Design sensitivity analysis usually computes the sensitivity of a solution with respect to design parameters, such as the thickness, height, width, etc., of a shell structure. On the other hand, in our interaction method, we compute the sensitivity with respect to the *cursor's screen position* $\mathbf{d}$. This is the point that distinguishes our design sensitivity from others. Typically, sensitivity analysis is used for optimization purposes.

The sensitivity is computed with respect to design parameters to find the combination of these parameters that maximizes an evaluation function under constraints. The optimization procedure is usually offline, and there is no user interaction. On the other hand, our system uses the sensitivity to accelerate the interaction. Our computational scheme based on design sensitivity allows the user to interact with the design while receiving instant simulation response, so that the shape can be optimized according to the user's own criteria.

In our approach, the user edits a design in a click-and-drag fashion. An element of the designed shape (such as a vertex, an edge, or a face) is selected by clicking and manipulated by dragging. Since the user directly edits the design vector $\mathbf{X}$ with the aid of a pointing device, the two-dimensional screen position $\mathbf{d} = (d_1, d_2)^T$ of the cursor is related to the design vector $\mathbf{X}$ during a dragging operation. Hence, during a dragging operation, the design vector is parameterized by the pointer's positions, expressed as $\mathbf{X}(\mathbf{d})$. We obtain the sensitivity modes with respect to the pointer's position in a manner analogous to sensitivity analysis with respect to the design parameters, slightly modifying Equation 5.10 to obtain Equation 5.15:

$$\mathbf{s}_i = -\left(\frac{\partial \mathbf{R}}{\partial \mathbf{x}}\right)^{-1}\left(\frac{\partial \mathbf{R}}{\partial \mathbf{X}}\right)\frac{\partial \mathbf{X}}{\partial d_i}. \tag{5.15}$$

The vector $\partial\mathbf{X}/\partial d_i$ represents the movement of the mesh vertices with respect to the horizontal or vertical movements of the cursor. The relationship between the mesh vertices and the position of the cursor can be specified arbitrarily. The best choices are discussed in detail in the next subsection.

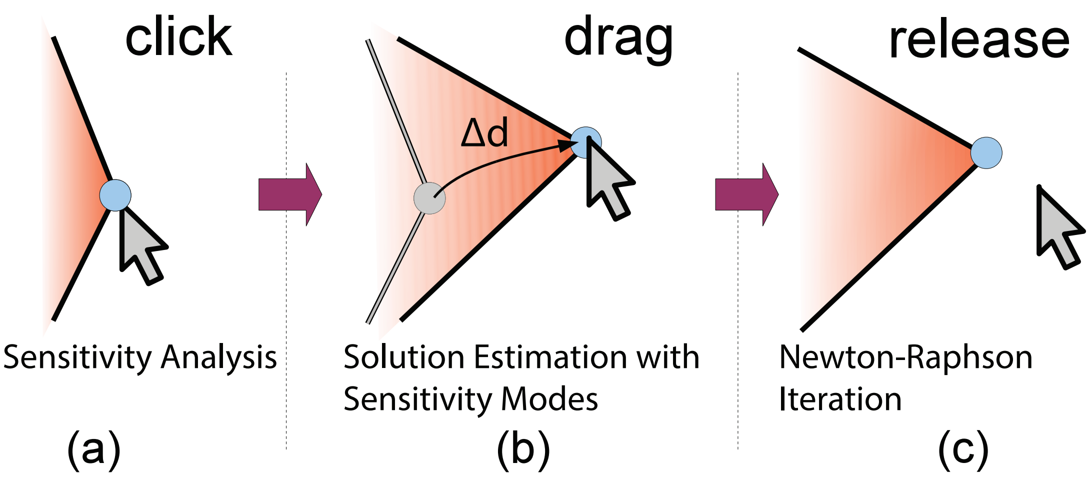


Figure 5.5: *Computations during each stage of a user's dragging operation.*

Figure 5.5 shows what kinds of computations are allocated during each stage of a user's dragging operation. When the user clicks the mouse button, the system computes the design sensitivity, $\mathbf{s}_1$ and $\mathbf{s}_2$, with respect to the cursor's horizontal and vertical positions, d_1 and d_2, using Equation 5.15. When the user drags the mouse, the system displays the approximate solution $\mathbf{x}$ from the sensitivity analysis in real time:

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{s}_1 \Delta d_1 + \mathbf{s}_2 \Delta d_2, \tag{5.16}$$

where the $\mathbf{x}_0$ is the solution stored when the mouse button is clicked, and Δd_1 and Δd_2 are the horizontal and vertical movements of the cursor from its location when the mouse button is clicked. Note that this computation only involves a linear combination of three vectors, and thus it is completed very quickly. Finally, when the mouse button is released, the system computes the actual static FEM solution via the Newton-Raphson iterations of Equation 5.4. In these iterations, the initial value is the design sensitivity approximation of Equation 5.16 where the mouse button is released.

In Figure 5.6, we illustrate how our sensitive interaction scheme differs from other methods in the case when $\mathbf{X}$, $\mathbf{R}$, and $\mathbf{x}$ are one-dimensional. In this figure, the design vector is changed from $\mathbf{X}_a$ to $\mathbf{X}_b$. FEM typically finds the static solution from scratch ($\mathbf{A}$ in the figure). *Responsive FEM* reuses the solution for the previous shape as an initial prediction ($\mathbf{B}$ in the figure). Our sensitive interaction scheme computes the solution starting from the further estimation of sensitivity analysis ($\mathbf{C}$ in the figure). By using this first-order approximation, the Newton-Raphson iterations converge more quickly than in a typical FEM simulation or responsive FEM.

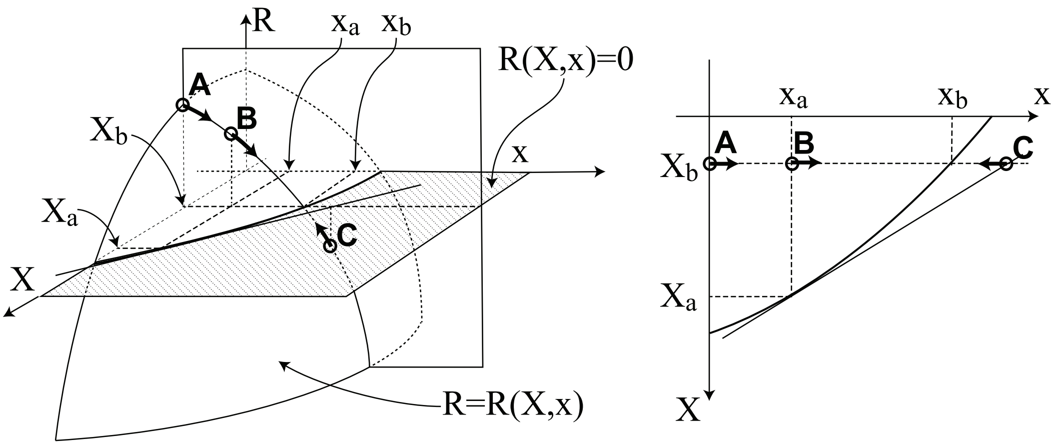


Figure 5.6: *A geometric comparison of the sensitive interaction scheme with other methods. The design vector is changed from $\mathbf{X}_a$ to $\mathbf{X}_b$. $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ represent the starting points of the Newton-Raphson iterations to obtain the solution $\mathbf{x}_b$ for the design vector $\mathbf{X}_b$ in typical FEM, responsive FEM, and sensitive interaction, respectively .*

5.1.4 Updating a Two-Dimensional Mesh

In this subsection, we explain how to change the positions of the mesh vertices $\mathbf{X}$ with respect to the position of the two-dimensional cursor $\mathbf{d}$ in the case of a two-dimensional mesh, expressed as $\mathbf{X}(\mathbf{d})$. This mesh update information is used in the sensitivity analysis of Equation 5.15, and FEM analysis is performed on the updated mesh during the user's interactive design. Thus, mesh updating must satisfy two fundamental properties: (i) linearity and (ii) small mesh distortion. We describe these two properties individually.

The first property (linearity) requires that the node vertices move linearly with respect to the cursor ' s movements. If the there is a high degree of nonlinearity between mesh vertex positions and the cursor ' s position, the first-order sensitivity modes computed in Equation 5.15 cannot provide a good estimate of the changes in the solution vector, since the first-order relationship $\partial\mathbf{X}/\partial\mathbf{d}$ in this equation is not a good estimator of mesh movement. Hence, we linearly update the ith vertex of the mesh, $\mathbf{X}_i$, as the cursor moves:

$$\Delta\mathbf{X}_i = \mathbf{\Psi}_i \Delta\mathbf{d}, \tag{5.17}$$

where $\mathbf{\Psi}_i$ is the 2×2 tensor relating cursor movement $\Delta\mathbf{d}$ to vertex movement (shown in Figure 5.7-*left* and -*right*). We define such a tensor for every vertex in the mesh. Then, $\mathbf{\Psi} = [\mathbf{\Psi}_1 \;\; \mathbf{\Psi}_2 \;\; \ldots, \mathbf{\Psi}_n]^T$, and the sensitivity is calculated from the linear relationship between the positions of the mesh vertices and the positions of the cursor in Equation 5.15:

$$\frac{\partial\mathbf{X}}{\partial d_i} = \mathbf{\Psi}\mathbf{e}_i, \tag{5.18}$$

where $\mathbf{e}_i$ is the ith unit vector in the screen coordinates. The Laplacian smoothing method is frequently applied to the updating of two- or three-dimensional meshes because it has the desirable property of keeping mesh distortion small. However, Laplacian smoothing is nonlinear, and thus is not suitable for our sensitive interaction scheme. Laplacian smoothing updates the vertex positions by assuming that all edges are springs and the vertices are masses, and solving for equilibrium in the resulting mass-spring system. Solving for equilibrium in a mass-spring system introduces nonlinearity, since the internal force of even a single spring is nonlinear with respect to its end point position in a two-dimensional system. Hence, Laplacian smoothing is not suitable for our purposes.

The second property (small mesh distortion) requires that mesh distortion remain small after mesh updating. The accuracy of an FEM simulation depends heavily on the mesh quality. Even one inverted or extremely distorted mesh usually results in very large errors in a static FEM simulation. Since we run a static FEM simulation on an updated mesh, uniform, well-shaped mesh elements must be maintained after mesh updating. $\mathbf{\Psi}$ must change smoothly throughout the mesh, and local extreme deformations must be avoided. In a typical design sensitivity analysis, where the sensitivity is used for offline optimization and not for interaction, infinitesimal mesh movement is assumed, and the map $\partial\mathbf{X}/\partial\mathbf{d}$ is used only for the sensitivity analysis and not for mesh updating. For such optimization purposes, we note that mesh updating is often limited to vertices on the boundary of the shape, while the nodes inside the shape are left intact [165]. When only the positions of boundary vertices are updated, elements near the boundary are quickly distorted (as shown in Figure 5.7-*right*), and would not be appropriate for our sensitive interaction scheme.

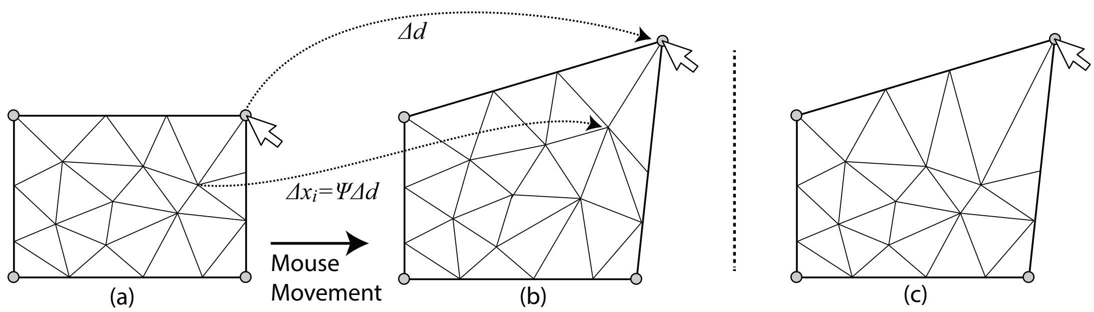


Figure 5.7: *Mesh updating from an original mesh (a) to an updated mesh (b). A mesh vertex* $\mathbf{x}_i$ *is updated linearly with a map* $\boldsymbol{\Psi}$ *according to the mouse movement* $\mathbf{d}$. *The typical mesh updating procedure used in sensitivity analysis moves only the nodes on the boundary, (c) inducing a distorted mesh, which is not desirable for FEM analysis.*

We assume the designed shape is represented by a polygon, and refer to the vertices of the polygon as control vertices (CVs). We adopt coordinate-based mesh manipulation to deform the initial mesh linearly, keeping the mesh deformation uniform and avoiding local extreme deformations. In other words, we generate a set of linear tensors $\boldsymbol{\Psi} = [\boldsymbol{\Psi}_1\ \boldsymbol{\Psi}_2, \ldots]^T$ for all the vertices, so that they are smoothly changed via coordinate-based interpolation inside the polygon. The system first defines the map $\boldsymbol{\Psi}$ on the CVs, then interpolates $\boldsymbol{\Psi}$ linearly on the edges, and finally interpolates $\boldsymbol{\Psi}$ inside the polygon, using coordinate-based interpolation.

$$\Psi_{vertex} \overset{\text{linear}}{\longrightarrow} \Psi_{edge} \overset{\text{PMVC}}{\longrightarrow} \Psi_{internal}. \tag{5.19}$$

First, we describe how cursor motion affects the control vertices (CVs). The system allows us to define an arbitrary map $\boldsymbol{\Phi}_i$ on the control vertices, which may vary according to context in the design process. Figure 5.8 shows some sample definitions of the map $\boldsymbol{\Phi}$ on the CVs. Typically, our tools use: (Figure 5.8-*left*) $\boldsymbol{\Psi} = \mathbf{I}$ or $\boldsymbol{\Psi} = \mathbf{0}$, where the CV follows the cursor or remains stationary, respectively; (Figure 5.8-*center*) $\boldsymbol{\Psi} = \mathbf{e}\mathbf{e}^T$, where the CV shadows the cursor movement only in the direction $\mathbf{e}$; or (Figure 5.8-*right*) $\boldsymbol{\Psi} = \mathbf{e}_i\mathbf{e}_j^T$, where the CV moves along $\mathbf{e}_i$ when the cursor moves along $\mathbf{e}_j$.

After we have defined $\boldsymbol{\Psi}$ at all the CVs, we interpolate $\boldsymbol{\Psi}$ in the remainder of the domain. Since $\boldsymbol{\Psi}$ is known at all boundary CVs, we linearly interpolate along the boundary and use *positive mean value coordinates* (PMVC) [102] to efficiently determine the tensor field $\boldsymbol{\Psi}$ throughout the domain. PMVC builds on the concept of *mean*

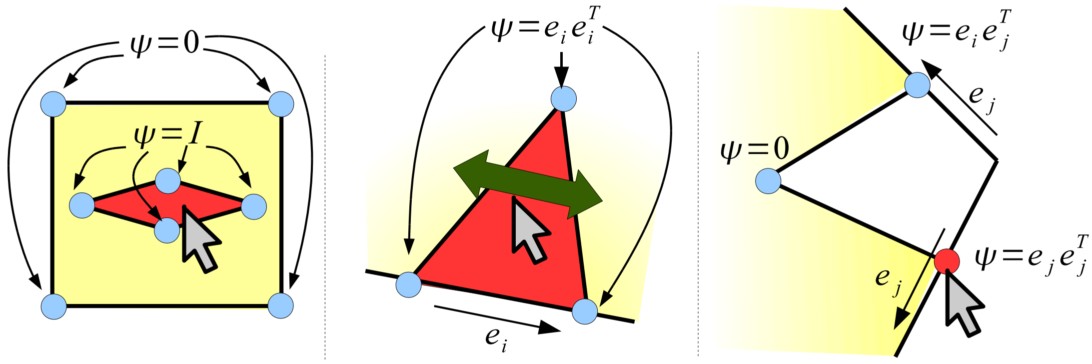


Figure 5.8: *Action of* $\mathbf{\Psi}$ *for (left) translation of an interior (diamond) dart, (center) translation of a boundary dart, and (right) adjustment of a boundary dart opening.*

value coordinates (MVC) [59] by incorporating the notion of *visibility*, which (we found) enhances the coordinates ' interpolation capability in highly concave shapes.

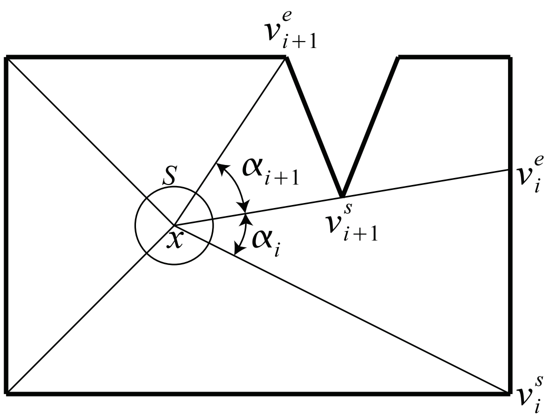


Figure 5.9: *A simple concave shape to illustrate the positive mean value coordinate (PMVC) method. The value at the point x is interpolated from the values on the boundary edges, where they are visible from the point x.*

Here we briefly explain the PMVC interpolation technique for mesh updating. Figure 5.9 illustrates an interpolation inside a simple shape, and we use this to explain the procedure. We consider a point x inside the shape, and compute the map $\mathbf{\Psi}$ at this point x by interpolating the values of $\mathbf{\Psi}$ on the boundary:

$$\Psi(x) = \frac{\int_{s \in S} \frac{\Psi(\pi(s))}{||\mathbf{x}-\pi(s)||} d\sigma}{\int_{s \in S} \frac{1}{||\mathbf{x}-\pi(s)||} d\sigma}, \tag{5.20}$$

where S is the unit circle centered at x, and $\pi(s)$ is the first point of intersection between the boundary and the half line directed by $s \in S$. To carry out the integration in Equation 5.20, we use the following analytical formula:

$$\int_{\gamma_i} \frac{f(\pi(s))}{||\mathbf{x} - \pi(s)||} ds = \tan\left(\frac{\alpha_i}{2}\right)\left(\frac{f(\mathbf{v}_\mathbf{i}^\mathbf{s})}{||\mathbf{v}_\mathbf{i}^\mathbf{s} - \mathbf{x}||} + \frac{f(\mathbf{v}_\mathbf{i}^\mathbf{e})}{||\mathbf{v}_\mathbf{i}^\mathbf{e} - \mathbf{x}||}\right), \tag{5.21}$$

where the line segment with endpoints $\mathbf{v}^\mathrm{s}$ and $\mathbf{v}^\mathrm{e}$ is on the boundary visible from the point *x*, α is the viewing angle subtended by the line segment (as depicted in Figure 5.9), and f utilizes either $\mathbf{\Psi}$ or 1 to compute the integral in Equation 5.20. More details can be found in the original paper [102].

Fast and slow remeshing. Over the course of multiple large manipulations, a mesh may be distorted so much that it may need remeshing. We first attempt *Delaunay smoothing*, updating mesh connectivity while retaining nodal positions. With our FEM discretization, displacement and sensitivity are stored at the vertices and need not be recomputed, making this an inexpensive way to improve the mesh. If the mesh quality (the ratio of the diameter of the incircle to the maximum edge length) remains undesirable, our system rebuilds the mesh from scratch and interpolates the simulation state to the new mesh, using barycentric coordinates. Figure 5.10 compares the two *linear* mesh deformation models we implemented, based on MVC and PMVC. As the figure indicates, PMVC produces a homogenous deformation that avoids distortions and inversions.

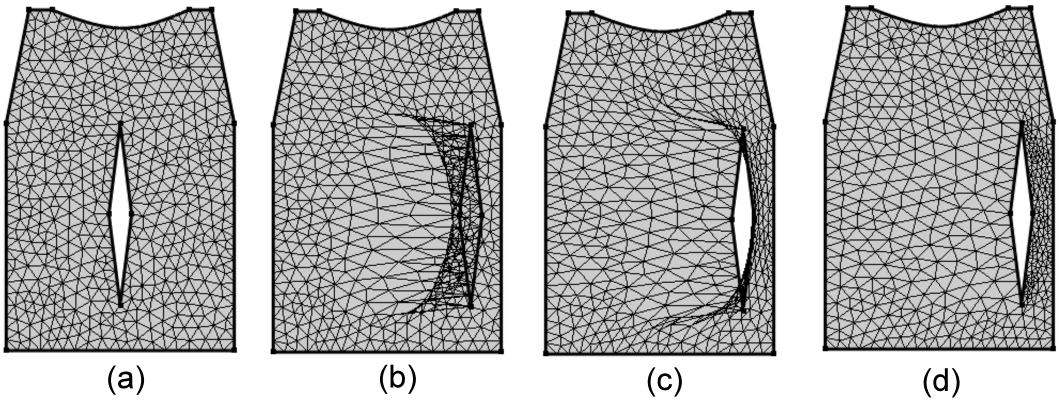


Figure 5.10: *Our two-dimensional pattern manipulation employs positive mean value coordinates (PMVC) with Delaunay smoothing. This allows for linear interpolation over a domain, while enabling more satisfactory interpolation in non-convex, higher-genus domains. The following figures illustrate the advantages: (a) an undeformed mesh, (b) mesh manipulation with MVC, (c) mesh manipulation with PMVC, and finally (d) PMVC with Delaunay smoothing.*

5.1.5 Augmentation of Sensitive Interaction for Nonlinear Response

In this subsection, we describe further enhancement of the previous *sensitive interaction* scheme, considering the nonlinearity of the simulation response with respect to design changes originating from cursor movement on the screen. While linear sensitivity is a good first step, we found that for the larger edits that are typical at the beginning of the design process, or in radical redesigns, the local model produced by sensitivity analysis (i.e., linearization of the pre-revised configuration) does not remain valid over the full extent of editing operations.

Typically, there is a high degree of nonlinearity in the relationship between the solution of a static FEM simulation and the shape being simulated. Thus, estimating the solution via linear sensitivity provides a good prediction only in limited local areas. We explain this with the simple cantilever deformation example shown in Figure 5.11. A force **F** is applied vertically at the tip of a cantilever beam with length l, width b, and thickness h. Assuming linearized infinitesimal deformation, the resulting deformation y at the tip of the cantilever beam is given by

$$y = -\frac{Fl^3}{3EI}, \quad I = \frac{bh^3}{12}, \tag{5.22}$$

where E is Young's modulus and I is the moment of inertia of the area [157]. In Equation 5.22, we observe that y varies linearly with respect to F, which is the assumption of linearized deformation. On the other hand, y is nonlinear with respect to the design parameters; it varies as the cube of l, the inverse of b, and the inverse cube of h. As this simple example illustrates, the simulation results can be highly nonlinear with respect to the design, and a linearized approximation often produces undesired results.

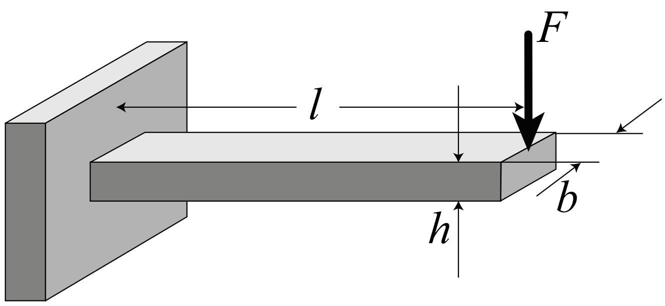


Figure 5.11: *Cantilever deformation.*

Moreover, in nonlinear deformation simulations, buckling, wrinkling and static friction all contribute to a nonlinear relationship between the rest shape and the deformed shape. In short, a linear sensitivity response centered about the original design is insufficient when the changes in design are large. Therefore, we extend the basic sensitivity-based approach by accounting for nonlinearity via *progressive nonlinear modeling*. During idle times, such as pauses in mouse movement, Newton-Raphson iterations are computed to obtain a convergent solution in real time. These iterations begin with the approximation produced via sensitivity-based interpolation. Once convergence is obtained, the value and its sensitivity are computed and stored in a cache to enhance the nonlinear approximation of the sensitivity-based prediction (see Figure 5.12).

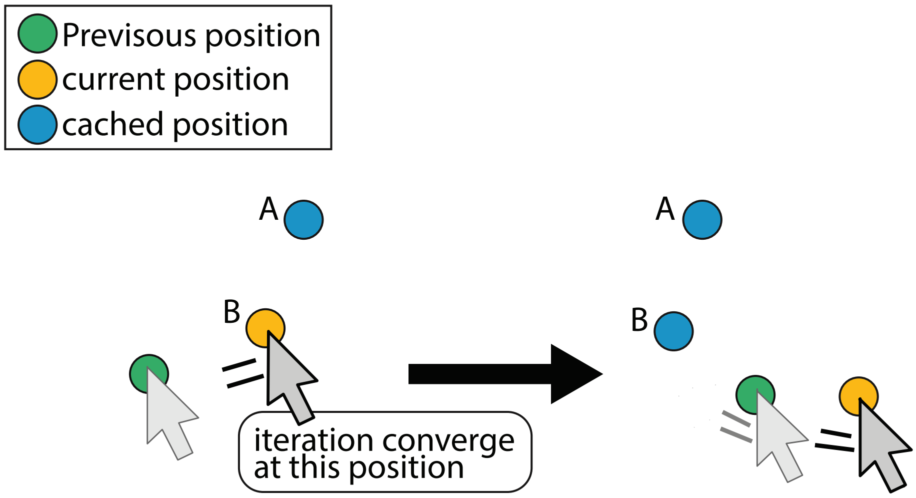


Figure 5.12: *As the mouse moves, the system continues to runs Newton-Raphson iterations, in addition to the approximation from sensitivity modes at the cached positions (blue), from the value at the previous cursor position (green). Once a convergent solution is obtained (at position B in the figure), the value and its sensitivity are computed and stored in a cache to enhance the nonlinear approximation.*

Progressive nonlinear modeling During the most crucial interactive editing operations—when the designer hesitates in selecting between multiple design alternatives— an operation may endure over a longer period (a few seconds), and pause at various points in the design space as they are being considered. In these cases, we take advantage of the available time to construct additional linearizations (sensitivity matrices), thus building up a nonlinear representation of the local behavior. Compared to up-front precomputation, this *progressive* enrichment of the local model allocates the computation in proportion to the interest in a given region of the design space (i.e., it is most accurate near parts of the design that interest the user).

Here we explain how we compute a nonlinear interpolation from cached solutions and their sensitivities at several locations. In the context of offline optimization, some authors have considered *moving least squares* (MLS) interpolation [32]. MLS requires dense, well-spaced samples, does not make use of derivative information, and breaks down when the sampling pattern degenerates (i.e., there are too few samples along one dimension). To alleviate these difficulties, Martin et al. [109] used a generalized moving least squares (GMLS) procedure, generalizing Hermite-style splines to the MLS context. Observing that GMLS makes direct use of sensitivity matrices [10, 62], we are motivated to adopt GMLS interpolation.

When the cursor moves to position $\mathbf{d} = (d_1, d_2)^T$, our system can draw upon two kinds of readily reusable data: the draped configuration $\mathbf{x}^0 \in \mathbb{R}^{3n}$ (i.e., zero-order data) at the previous cursor position $\mathbf{d}^0 \in \mathbb{R}^2$, and the sensitivity $\mathbf{s}_1^m = \partial\mathbf{x}/\partial d_1, \mathbf{s}_2^m = \partial\mathbf{x}/\partial d_2$ of the draped configuration $\mathbf{x}^m \in \mathbb{R}^{3n}$ (i.e., first-order data) at the cached previous configurations $\mathbf{d}^m \in \mathbb{R}^2$ $(m = 1, \ldots, M)$. Since GMLS is a direct extension of MLS, it is possible to extend the typical derivations of MLS and GMLS interpolation to account for a combination of zero- and first-order samples. In particular, following the notation of Martin et al. [109], the interpolated displacement field is given by $\mathbf{x}(\mathbf{d}) = \mathbf{a}(\mathbf{d})\mathbf{p}(\mathbf{d}) \in \mathbb{R}^{3n}$, where $\mathbf{a} \in \mathbb{R}^{3n\times 3}$ is a coefficient matrix applied to the monomial vector $\mathbf{p} = (1, d_1, d_2)^T$. At a given cursor position $\mathbf{d}$, the coefficients $\mathbf{a}$ are the minimizers of the least-squares error metric:

$$
\begin{aligned}
J(\mathbf{a}) = &\sum_{m=0}^{M} w(\mathbf{d}-\mathbf{d}^m)\,\|\mathbf{a}\mathbf{p}^m - \mathbf{x}^m\|^2 \\
&+ \sum_{m=1}^{M} w(\mathbf{d}-\mathbf{d}^m)\sum_{j=1}^{2}\left\|\mathbf{a}\frac{\partial\mathbf{p}}{\partial d_j} - \mathbf{s}_j^m\right\|^2,
\end{aligned}
$$

where the weighting function $w(\mathbf{d}-\mathbf{d}^m) = 1/(\|\mathbf{d}^m - \mathbf{d}\|^2 + \epsilon^2)$, and ϵ is a small constant that keeps the weight finite (we use $\epsilon^2 = 10^{-3}$). As in the usual (G)MLS derivation, we analytically minimize J with respect to $\mathbf{a}$, and obtain:

$$
\begin{aligned}
\mathbf{x} &= \sum_{m=0}^{M}\mathbf{x}^m N^m(\mathbf{d}) + \sum_{m=1}^{M}\sum_{j=1}^{2}\mathbf{s}_j^m N_j^m(\mathbf{d}), \\
N^m(\mathbf{d}) &= \mathbf{p}(\mathbf{d})^T\mathbf{G}(\mathbf{d})^{-1}\mathbf{p}(\mathbf{d}^m)w(\mathbf{d}-\mathbf{d}^m)\ , \\
N_j^m(\mathbf{d}) &= \mathbf{p}(\mathbf{d})^T\mathbf{G}(\mathbf{d})^{-1}\frac{\partial\mathbf{p}}{\partial d_j}w(\mathbf{d}-\mathbf{d}^m)\ , \\
\mathbf{G}(\mathbf{d}) &= \sum_{m=0}^{M} w(\mathbf{d}-\mathbf{d}^m)\mathbf{p}(\mathbf{d}^m)\mathbf{p}(\mathbf{d}^m)^T \\
&\quad + \sum_{m=1}^{M} w(\mathbf{d}-\mathbf{d}^m)\sum_{j=1}^{2}\frac{\partial\mathbf{p}}{\partial d_j}\frac{\partial\mathbf{p}}{\partial d_j}^T\ .
\end{aligned}
$$

5.2 Sensitive Couture: Interactive Clothing Pattern Authoring System

We applied the algorithm described in Section 5.1 to the interactive clothing pattern system called Sensitive Couture (SC). SC presents the 2D clothing pattern and 3D clothing drape simultaneously. This section describes the interface of our clothing pattern design system from the viewpoint of the users. All selection operations feel naturally at home in either the 2D or 3D window.

A typical pattern design session begins with layout of a “blank canvas.” While SC affords from-scratch creation of patterns, designers typically begin with a *sloper*

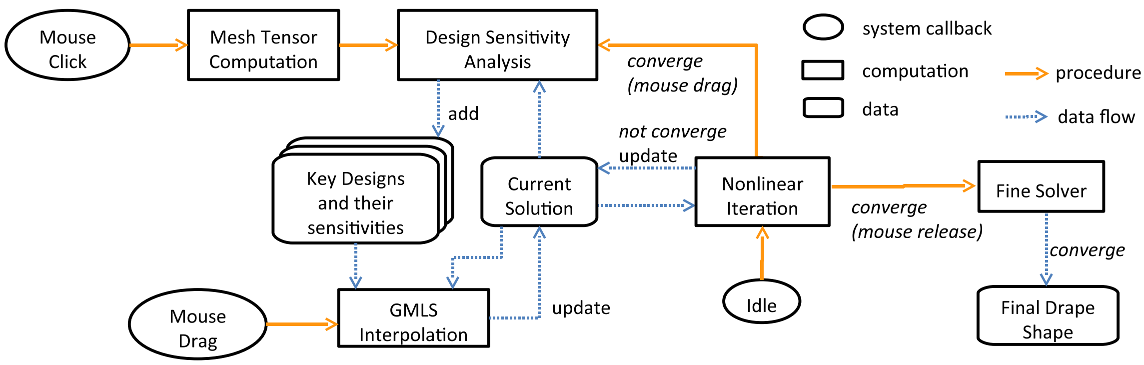


Figure 5.13: *Our full system flowchart.*

(or *block*), a basic pattern drafted to standard measurements intended as a generic starting point [52] (see Figure 5.20). Slopers are typically *parameterized* by height, girth, sleeve length, and so forth.

Whereas typical CAD tools might navigate the parametric space using sliders, SC further allows the designer a "tangible" navigation of parameter space via direct manipulation of the 2D pattern and the 3D draped shape, reinforcing the idea that all components are synchronized and can be accessed in any order.

Indeed, every aspect of the SC user experience emphasizes a free-flowing (as opposed to sequential or "linear") design process. For instance, as the designer makes detailed alterations ("strokes") to patterns by inserting darts, modifying boundary curves, etc., these strokes "ride" over the sloper, in the sense that the designer may revisit the parameters of the sloper at any time without undoing the creative, style-defining strokes.

If the sloper serves as the blank canvas, then the rich of diversity of final designs comes from the various creative strokes applied by the designer. Our research goal was to understand the under-the-hood infrastructure needed for interactive design in a real-application (practical) context, therefore we focused on a small but useful set of front-end tools intended to stress-test the underlying simulation infrastructure:

Curve edits. The designer may alter the shape and position of pattern boundary, which is defined by the control degrees of freedom (DOFs) of a spline. Positions are stored relative to the sloper, and are therefore naturally maintained over adjustment to the underlying sloper's dimensions.

Darts. The designer may add or modify *darts,* triangular folds that induce intrinsic curvature (so-called *cone singularities*), to make the clothing fit to a 3D body (see Figure 5.2). SC understands darts as "first-order primitives," so they have dart-specific DOFs to control their position, shape, and size. The designer adds a dart by drawing a line using the dart pen. If the line intersects a boundary, SC creates a triangular dart that “rides” the boundary, i.e., the designer may later freely slide the dart along the boundary.

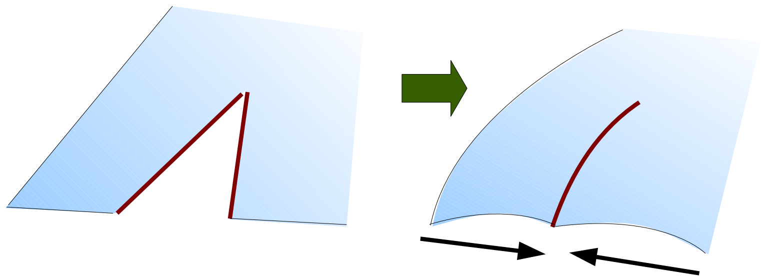


Figure 5.14: *Illustration of a typical dart.*

Sewing/pleating. The designer can specify that two boundary segments should be sewn. When the two segments differ in a length, a *pleat* is formed, i.e., a sequence of attractive doubled-back folds that gather a longer piece of fabric into a shorter length. In all cases, SC automatically selects the boundary orientations that avoids cloth inversion.

Symmetry. The designer may mark boundary pieces as symmetric about an axis, and subsequently SC enforces these symmetries.

Sloper parameters. We emphasize that the designer can adjust sloper parameters (e.g., sleeve length, waist width) at any time. These adjustments can be made by manipulating a slider, or by direct manipulation in the 3D view. For example, the user can “tug” on a skirt to make the skirt longer (see Figure 5.15).

5.3 Implementation of Sensitive Couture

In this section, we describe detailed implementation techniques specially required for clothing pattern design interface shown in the previous section. The interaction algorithm is mainly based on the nonlinear argumented sensitivity analysis shown in Section 5.1. Therefore, here we focus on a fast time integration scheme for static clothing simulation, a physical model of clothing we applied, and bidirectional editing algorithm.

Figure 5.15 illustrates the bidirectional Sensitive Couture (SC) workflow for designing garments on 3D models. SC enables designer edits of both the 2D patterns (see Figure 5.15-(b)) *and* the 3D draped model (see Figure 5.15-(a)).

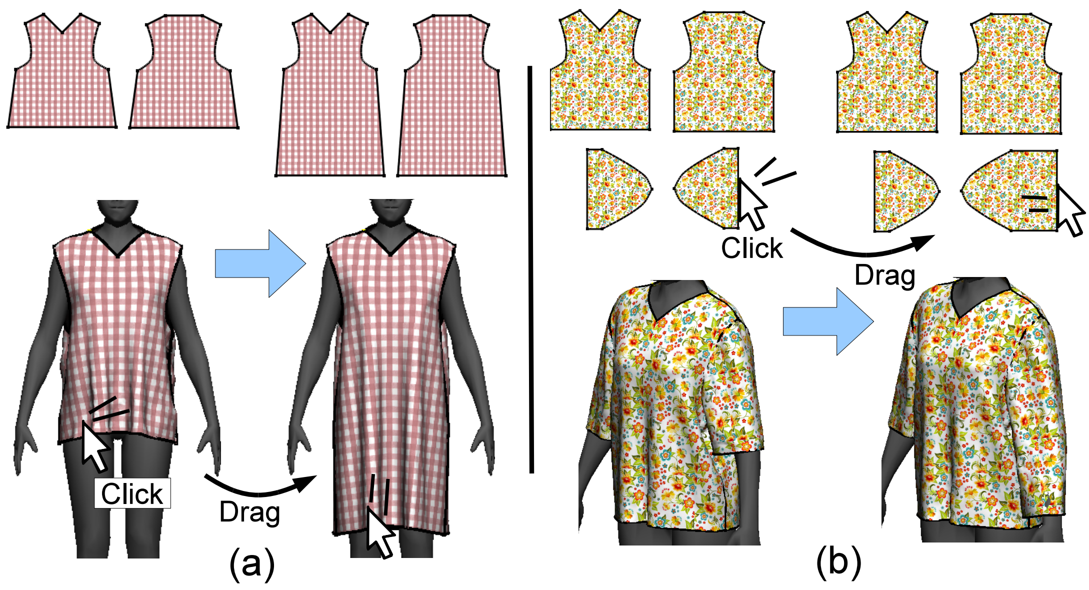


Figure 5.15: *Our system allow the user to manipulate 2D cloth pattern (b) or 3D cloth directory (a) with click and drag operation.*

5.3.1 Time Integration

Since SC displays the static drape of the cloth, any computation invested in capturing dynamics is a wasted effort. When a previous static solution is not available for an incremental sensitive update, SC must find the static equilibrium from scratch, and do so quickly. A from-scratch solution is necessary not only for the initial drape, but also when new 2D pattern elements are added midway through the design process.

One of the fastest ways we found for solving the static equilibrium (Equation 5.3) from scratch employs *kinetic damping* [17, 168], which integrates the (undamped) equations of motion while monitoring the total kinetic energy at each time step. When the kinetic energy reaches a local maximum (a condition evaluated by considering three consecutive time steps), the kinetic damping approach zeros the velocity (i.e., the kinetic energy). The intuition behind this is that in a conservative oscillatory system, when kinetic energy is at a maximum, potential energy is at a minimum. When the system is far from the minimal potential, damping slows the approach to the minimum, and when the system is close to the minimal potential, reducing the momentum helps to avoid overshooting the minimum. We found that kinetic damping is simple to implement in practice, and reaches the draped configuration faster than a dynamic simulation with Rayleigh damping, an application of gradient descent, or the stabilized Newton method.

We apply kinetic damping to the semi-implicit time-integration scheme proposed by Baraff and Witkin [11]. Since the coefficient matrices of the dynamic simulation and the sensitivity analysis are both positive definite, we solve the system using conjugate gradients preconditioned with ILU(0) [141]. While the IBM bending model has a constant Hessian, the StVK CST membrane model does not, and we must numerically factorize the matrix at every time step.

5.3.2 Physical Model of Cloth

In selecting a cloth model for an interactive tool, our primary desiderata were stability and fast computation. As with most cloth models, we treat bending and stretching separately [11, 34]. We have chosen to work with triangular meshes, but the overall framework does not depend on this choice.

Bending. Our final implementation uses the *isometric bending model* (IBM) of Bergou et al. [23], which has a constant energy Hessian (force Jacobian), and thus (a) eliminates the cost of force Jacobian computation in the implicit time integration, (b) provides a simple matrix-vector multiplication for the bending force computation (which would be easy to port to the GPU), and (c) ensures that the Hessian remains positive semi-definite for all configurations, thereby stabilizing the numerics. Before adopting IBM, we implemented a co-rotational treatment of the *discrete Kirchoff triangle* [19, 90], but found that we needed a stronger guarantee of stability in the context of our interactive tool. We then adopted nonlinear hinges [34, 71], which takes away some of the meshing independence of DKT [19], but increases the stability. Ultimately, the cost of force Jacobian computations in nonlinear models drove us to use IBM, which drops the generality of an arbitrary stress-strain map in favor of increased stability and speedier computations.

Membrane. Our final implementation uses a *stabilized* St. Venant-Kirchhoff (StVK) *constant-strain triangle* (CST). The usual StVK CST element can induce instability when compressed [169], because its force Jacobian becomes indefinite. As in the work of Teran et al. [154], when an element is in a compressed configuration, we adjust the Jacobian entries to eliminate negative eigenvalues. This stabilization affects only the *trajectory* toward the draped configuration, and does not alter the set of solutions to the static equilibrium equations. In summary, this stabilization assures stability without affecting the draped shape. We first tried a geometric model governed by changes in edge length and triangle area [71], but found that the area term induced instabilities. The length term alone was stable, but lacked similarity to fabric [50]. We therefore transitioned to the CST model, first using a co-rotational treatment [25], which was stable, but was relatively slow in converging to the equilibrium draped-shape StVK membrane, compared to our final (quartic) CST element. In hindsight, we might also have applied the stabilization to the area term of the geometric model, but since the stabilized StVK CST worked well and provided a convergent model, we were satisfied. The details of this algorithm are presented in Appendix B.

Robustness evaluation We have found the combination of the guaranteed positive semi-definite IBM bending model and the stabilized StVK CST membrane model to be more stable than other combinations in the ongoing development of SC. Since, to our knowledge, this is the first reported attempt to combine these approaches, we include (for reference purposes) the results of two stability benchmarks (see Figure 5.16), in which static equilibrium (Equation 5.3) is solved via kinetic damping (see Section 5.3.1), starting from a random initial estimate. The alternatives we tried previously were either unstable or were slower to converge.

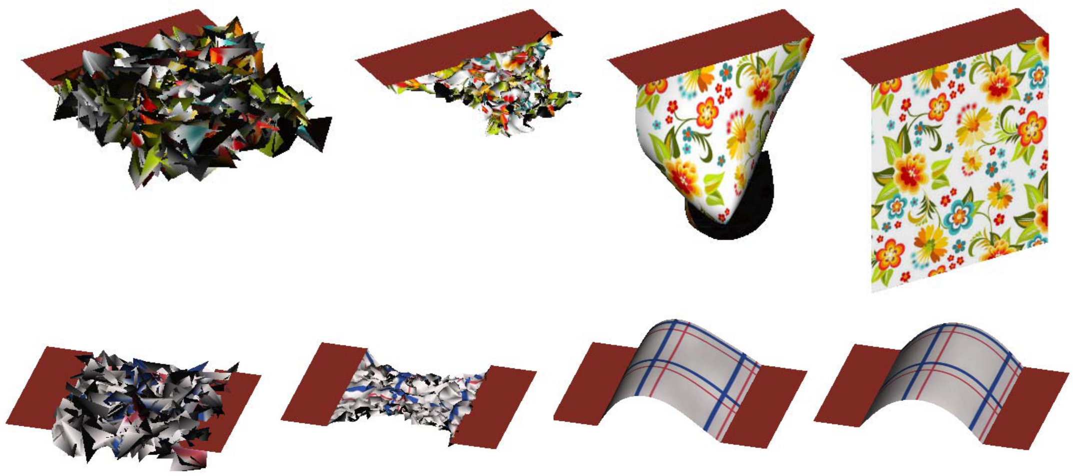


Figure 5.16: *Stability Benchmark: (top) Snapshots from the initial, third, 50th, and 200th iterations for the draping of a rectangular (membrane-dominated) textile (3K triangles, total computation time of 12 s). (bottom) Snapshots from the initial, third, 50th, and 100th iterations for the static equilibrium of a rectangular (bending-dominated) thin plate with a bending stiffness 100× that of a textile (2K triangles, total computation time of 4 s).*

Contact model SC models the contact between a garment and a body, but does not consider garment self-contact. Contact and friction between body and cloth are essential for a cloth-draping simulation. We opted for speed and robustness at some expense to accuracy. As with seaming, we enforce contact constraints at mesh nodes using penalty springs. For contact, we place normal springs at collision sites detected by a one-time, precomputed, adaptive, signed distance field [63]. We model friction using a moving anchor spring method [54] that enables both static and dynamic friction modes. Contacting nodes are connected by springs to seeded anchor vertices placed on the contacted surface. We then update (or release) the anchor positions with respect to nodal movement, to ensure that Coulomb's Law is satisfied by these anchor spring forces. This allows us to obtain stable draping and frictional wrinkling.

Seams Surprisingly, the performance of ILU(0) preconditioning is significantly influenced by the treatment of *seams*. SC sews the boundaries of corresponding panels using Hookean springs. In general, the boundaries do not correspond in length (an important feature in dressmaking, used to create pleats and ruffles) or in connectivity. Therefore, SC connects the ("emitting") vertices of one panel with springs anchored at the ("receiving") boundary edges of another panel (see Figure 5.17). To avoid gaps at the seams, the seam springs are much stiffer than the textile tensile stiffness. Since the resulting linear system has both stiff and weak components, the success of ILU(0) reconditioning depends on the permutation of the matrix entries [141]. In a nutshell, ILU(0) favors permutations in which large entries appear earlier (toward the top left corner of the matrix) and small entries appear later (toward the bottom right corner). Entries associated with emitting vertices dominate entries associated with receiving vertices, which in turn dominate all other vertices.

Before considering penalty-based seams, we used Lagrange multipliers, which allow exact constraint enforcement without preselecting the spring stiffness. However, we were not satisfied with the increased size of the linear system and the indefiniteness of the resulting system matrix, the combination of which slowed the convergence. We then observed that the usual problems plaguing penalty methods are less relevant in our context. Since our problem setting is fixed (the spatial scales of cloth designs do not vary by orders of magnitude), a single initial estimate of the penalty stiffness is sufficient to obtain a good seal at the seams. The penalty method maintains the positive definiteness of the system, and since the set of dominant matrix entries is obtained by a simple tallying of seam vertices (a set which remains constant except during exceptional stitching events), the permutation of ILU(0) has negligible implementation and computational costs.

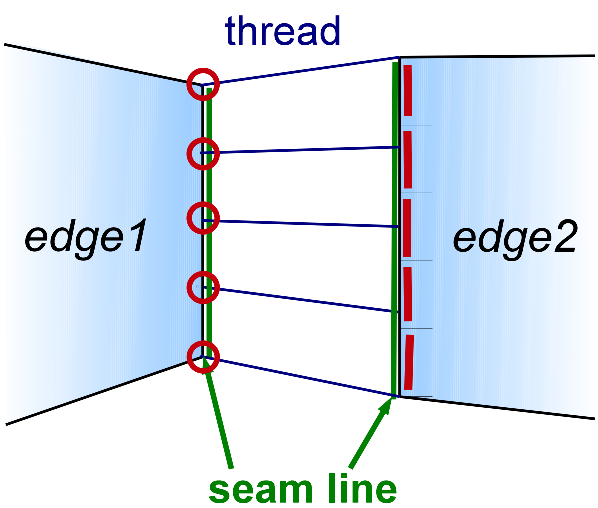


Figure 5.17: *A seam is implemented using penalty springs that connect nodes to elements on the seam lines.*

5.3.3 Sensitivity Analysis of Clothing

Full rank of the sensitivity matrix In general, for a nonlinear cloth statics problem, there exist configurations for which the stiffness matrix is close to singular. Consider a horizontal cantilevered rectangle of sufficiently thin fabric, which clearly cannot hold itself up. Quantitatively, gravity acts against the bending mode. The bending membrane stiffness ratio of a thin plate is $O(h^2)$, where h is the thickness measured relative to the object's characteristic length [108]. In principle, one could choose h to be so small that the configuration is nearly singular, with a kernel corresponding to normal displacements. In practice, we did not encounter this problem while using SC. However, for the sake of completeness, we did experiment with the cantilevered rectangle to induce this problem. Since the kernel must correspond to normal displacements, the remedy is straightforward: if the solver fails to converge, we re-compute S with a bending stiffness corresponding to $h = 1/100$, which is sufficiently thick to avoid a singular matrix. This modification of the bending stiffness is localized to the sensitivity computation, and does not affect the bending stiffness used in determining the ultimate static drape. We found that this stabilization was not invoked in any examples of clothing design.

Comparing different strategies Consider, as a canonical example, a hanging cloth partly draped over a sphere, and an editing operation in which the (undeformed, 2D material-space) length of the strip of cloth is lengthened. Figure 5.18 and the accompanying video compare the results of an editing operation using only a dynamic, kinetically-damped simulation augmented with linear sensitivity analysis, or augmented with progressive GMLS modeling. Here, Figures 5.18 (a) and (b) show the cached solutions employed by the sensitivity analysis. Observe (left to right in Figure 5.18) that the dynamic simulation lags behind the user's editing. The formation of distracting wrinkling artifacts can be understood by interpreting the dynamic simulation as a process that approaches the final equilibrium state in a fine-to-coarse manner. Linear sensitivity analysis, which can be viewed as a "global" or "implicit" approach, eliminates these artifacts. However, the linear model does not approximate the overall draped shape well. The GMLS interpolation exhibits stable results that are in better agreement with ground truth.

5.3.4 Bidirectional Sensitivity Enables Direct 3D Editing

Sensitivity information also enables SC to interpret editing operations applied directly to a 3D form. Suppose a slope parameter $g \in \mathbb{R}$ (e.g., sleeve length) is edited, as depicted in Figure 5.15a. If the left mouse button is depressed with the cursor over the 3D view, SC identifies the corresponding material point $(u, v) \in \mathbb{R}^2$ on the cloth, and computes the sensitivity vector $\mathrm{s} = \nabla_g \mathbf{x}(u, v) \in \mathbb{R}^3$ (i.e., the first-order 3D motion of the cloth at the chosen point with respect to $g \in \mathbb{R}$).

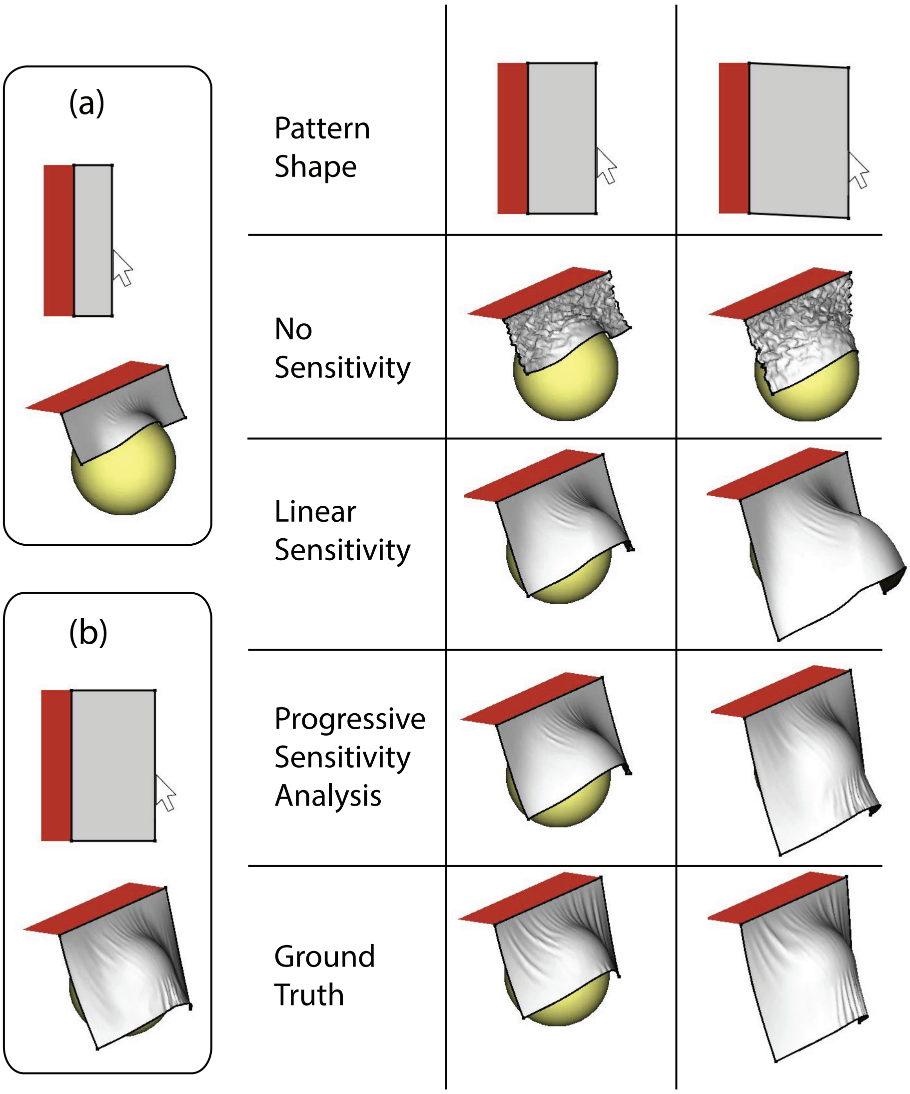


Figure 5.18: *Comparison of sensitivity strategies. Cloth is draped on a sphere with (from left to right) no sensitivity analysis, linear sensitivity analysis, progressive sensitivity analysis with GMLS, and nonlinear ("ground truth") static equilibrium. Here (a) and (b) show the cached solutions employed by the sensitivity analysis. The pure dynamic simulation lags behind the user's editing, and distracting wrinkling artifacts are formed. Linear sensitivity analysis eliminates many of these artifacts, but does not capture the draped shape, while GMLS interpolation exhibits stable results that are in better agreement with ground truth.*

The 3D vector $\mathbf{s}$ is projected onto the screen-space vector $\hat{\mathbf{s}} \in \mathbb{R}^2$, which gives the first-order motion of the chosen screen point with respect to g. As the user drags the cursor from $\mathbf{d} \in \mathbb{R}^2$ to $\mathbf{d} + \Delta\mathbf{d} \in \mathbb{R}^2$, SC updates the slope parameter via the incremental relationship

$$\Delta g = \hat{\mathbf{s}} \cdot \Delta\mathbf{d} / \|\hat{\mathbf{s}}\|^2.$$

Observe that when $\|\hat{\mathbf{s}}\|$ is small, the selected screen point is insensitive to g (for example, the position of a shirt collar may be independent of the sleeve length). Thus, for small $\|\hat{\mathbf{s}}\|$, we neglect the drag. A simple and useful extension would be to visualize the degree of sensitivity of a screen point before the left mouse button is depressed, as a guide for the user.

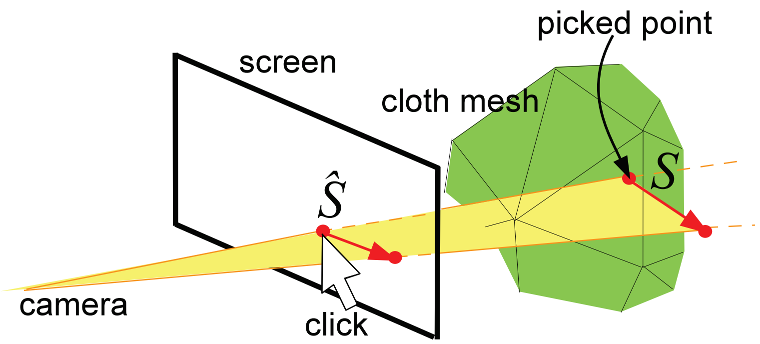


Figure 5.19: *Implementation of 3D Drag The sensitivity of the solution is projected on the screen.*

5.3.5 Progressive Refinement

We find that as users explore the design space with SC, detailed cloth behavior is unnecessary during the sketch-like exploratory phase, when design choices often change rapidly. On the other hand, during the slower, refinement-oriented phases of garment design, fine details are critical. We therefore focus on *progressive refinement*, a technique long-used in graphical applications to provide a sense of immediacy [77]. We employ the so-called *Cascading Multigrid* approach [31]. As users provide input, the drape is initially solved using a coarsened mesh. If we reach convergence at the coarse level *prior*, at the initiation of new design edits, we then warm start our fine-mesh solution with the coarse solution. In the preview, we always display the fine-mesh representation. Fine-level nodes are updated progressively, first via barycentric coordinates from the coarse solution, and later (when not interrupted) via direct updates from the fine-level solution.

5.4 Results and Discussion

Figure 5.20 shows various clothing designed for different characters starting from standard pattern templates. The sensitive couture allow the user to design a clothing that fits to the various target body shapes.

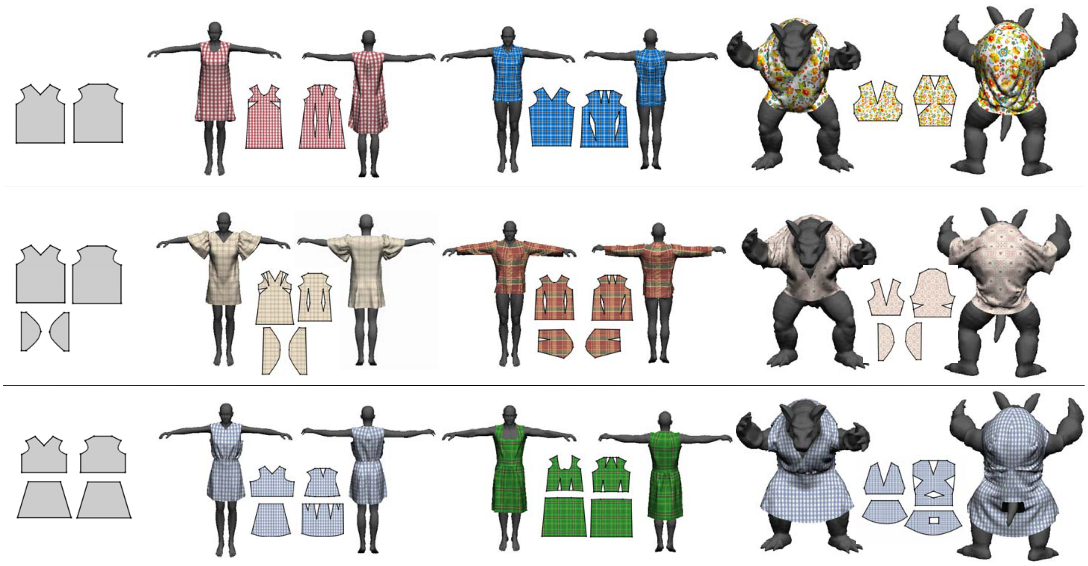


Figure 5.20: *Starting with just a few standard sloper templates artists can easily generate a wide variety of garments.*

Material parameters for the fabrics used are estimated both measurement and experiment. First, we measure arial densities of fabric and then we adjust stiffnesses so that simulations agree with real experiments on a draping on sphere problem. Figure 5.21 shows actual drapes on the sphere (basketball) and simulations of drape where stiffness of the cloth is manually adjusted so that the appearance of the drape is as similar as possible. contact parameters are inferred from the literature.

To achieve high-fidelity interactive rate simulation and editing tools we tried range of methods. In many cases we discovered that many techniques, that performed well in standard settings, did not satisfy our needs for seamless, interactive rate updates. Often obtaining working solutions required unexpected and complex combinations of methods. Throughout our discussion we have documented both *what did not work*, as well as what did (and thus made SC successful). Correspondingly we have provided comparisons throughout our discussion to explain these issues. These comparisons and validations can be found in Figures 5.18, 5.16, and 5.10.

Table 5.1 gives the simulation setting and runtimes for several design examples shown in this paper. Observe that speeds in the shape editing range from about 10 frames per second to about 22 frames per second. The sensitivity analysis takes about from 0.1 second to 0.5 second in the cloth examples and takes about from 0.4 second to about 0.8 second in the paper examples (shown in Figure 5.24).

Figure 5.21: *Adjustment of cloth's stiffness parameters.*

Here we wish to focus on how these parts sum to a whole in SC. As such, we have documented and captured (see our accompanying movies) the editing process and work flow of a variety of skilled and amateur clothing designers as they use SC to design a wide range of garments (see Figure 5.23). Garments designed in our SC sessions were then manufactured from the generated patterns. Figure 5.22 shows an actual pattern and an sewing machine used in our manufacture. See Figure 5.23 for snapshots of the editing process, the completed 2D designs, simulated final drapes, and corresponding manufactured garments.

Table 5.1: *The list of garments performance was measured and their mesh resolutions, material parameters, runtimes, and timings. All runtimes and timings are obtained on a laptop machine with Intel® Core™2 Duo 2.66GHz CPU with 4Gb of memory.*

		Fig. 5.20 bottom Armadillo	Fig. 5.23 Man	Fig. 5.23 Armadillo	5.24 Paper
coarse #nodes	min	1657	1583	1584	1572
	max	1697	1671	1638	1741
	ave	1671	1636	1610	1663
fine #nodes		10168	10149	10073	5032
bending stiff. (Nm)		1.0e-8	1.0e-8	5.0e-6	3.0e-4
stretch stiff. (N/m)		10	10	20	2000
areal density (kg/m^2)		0.2	0.2	0.15	0.08
during drag (FPS)	min	10.1	11.1	17.0	19.3
	max	19.9	17.2	22.0	22.2
	ave	17.0	15.0	19.3	20.7
coarse dynamic (FPS)		17.4	20.4	25.1	24.2
fine dynamic (FPS)		2.4	2.4	3.1	7.1
sensitivity analysis (ms)	min	82	170	116	419
	max	178	492	513	767
	ave	145	316	228	639
GMLS interpolation (ms)		0.38	0.36	0.4	0.39
remeshing (ms)		233	227	146	123

(a) (b)

Figure 5.22: *A printed pattern (a) and a sewing machine (b) used in our manufacture.*

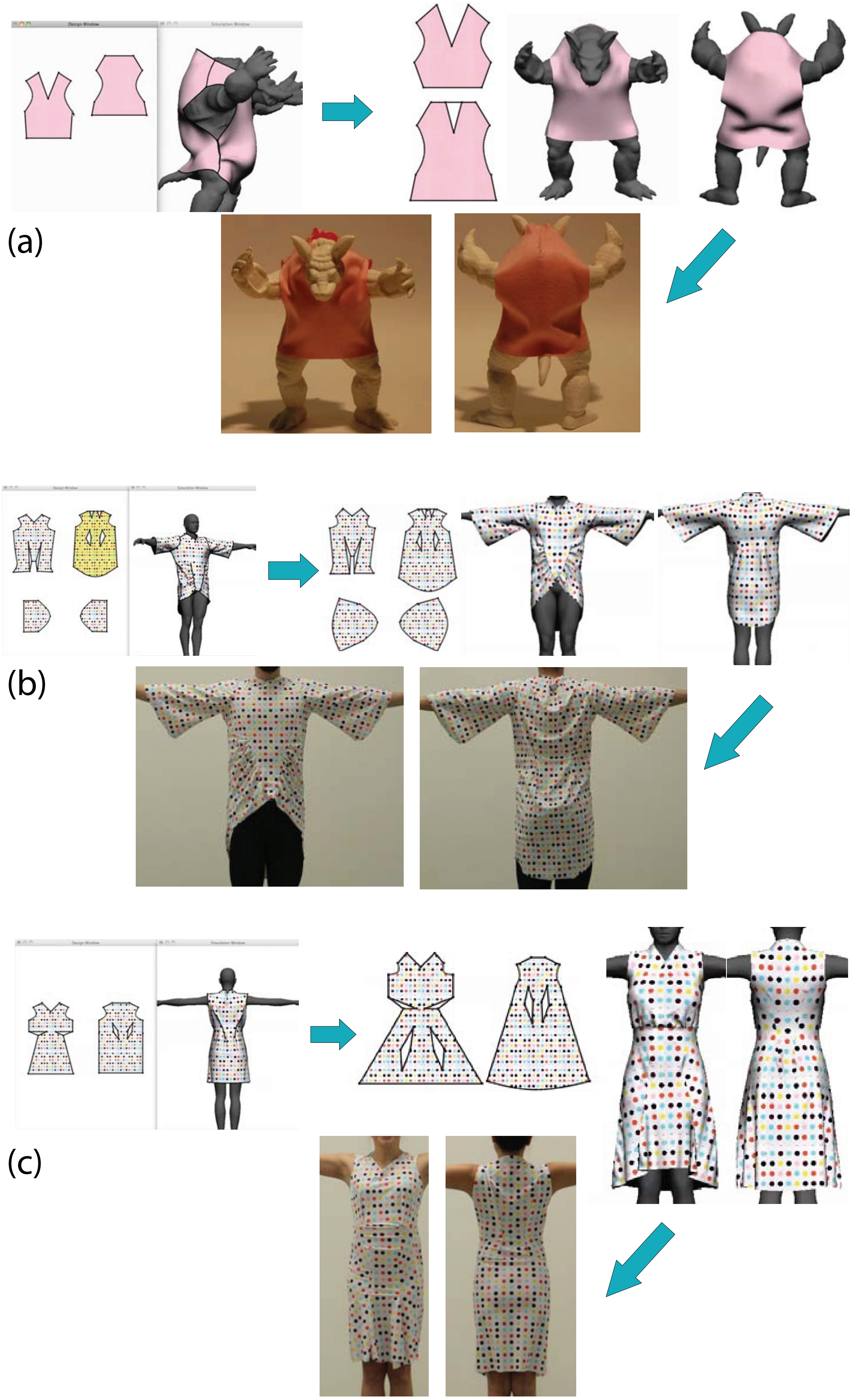


Figure 5.23: *The SC workflow: an interactive design session leads to a final 2D pattern and corresponding 3D drape pose which is then realized from the pattern.*

5.5 Limiations and Future Work

We have presented a novel, interactive garment design tool, Sensitive Couture (SC), that, for the first time, offers seamless bidirectional design and editing capabilities for the generation of 2D patterns and the online simulation of drape. SC provides a continuous, interactive, natural design modality in which the 2D design and 3D draped form receive equal status, are simultaneously visible, and seamlessly maintain correspondence. As such, artists are enabled to interactively edit and explore 2D designs and immediately observe how these changes affect 3D form.

While contact remains well-resolved for body-garment interaction, complex folding patterns will additionally require self-contact resolution. We are interested in exploring local methods of identifying and robustly treating self-contact and intersection in the SC framework. Similarly, as in the traditional couture setting, we currently consider the drape of SC garments on static 3D models. Clearly a desirable and challenging extension is for a dynamic preview capability allowing the simulation of 3D garment behavior subject to 3D model motion.

The design of shell-based objects also has exciting applications beyond garment design (see Figure 5.24). We expect that SC-type tools can be extended to address interactive design needs in architecture, industrial design, and engineering, e.g., tensile structures, metal folding processes, upholstery, balloons.

In principle, the 3D editing of more than two design degrees of freedom (DOFs) could follow the same line of thought: so long as no more than two of the DOFs have non-negligible screen space vectors at the drag point, the editing operation is unambiguous and straightforward to implement. However, the likelihood that the edit operation remains unambiguous decreases as the number of available DOFs increases, raising interesting directions for future work: (a) SC should intuitively convey to the user which DOFs are affected by the edit (information that varies over screen space), (b) is there an intuitive (and unambiguous, deterministic) way of mapping the 2D drag to simultaneous revision of more than two DOFs?

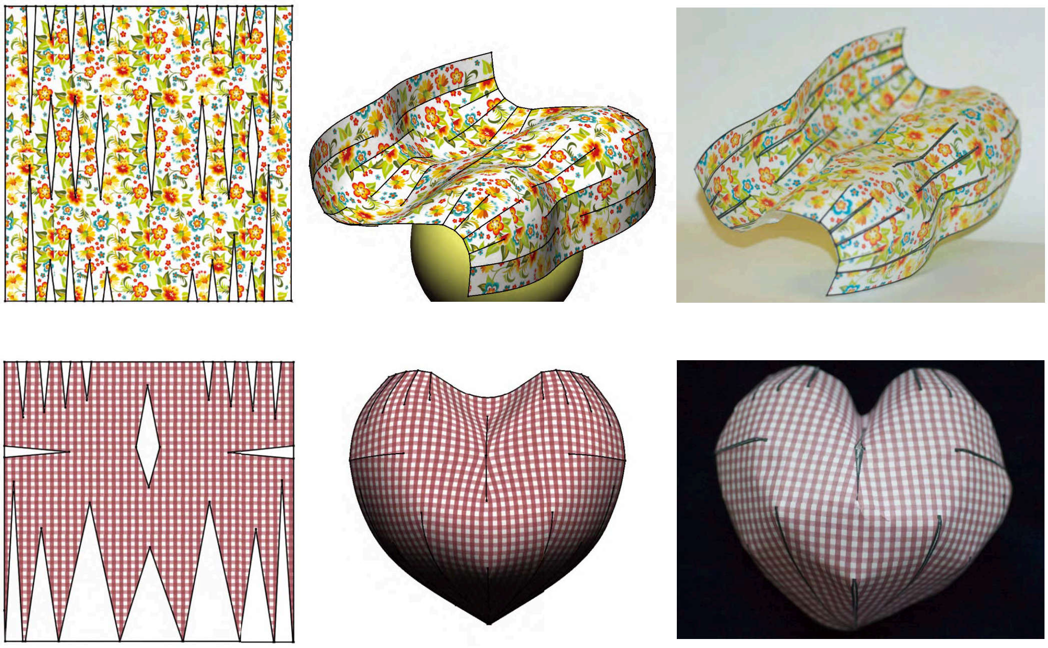


Figure 5.24: *In principle,* Sensitive Couture *could be extended to many other flexible materials, enabling the design of upholstery, alumnium sculptures, or these paper sculptures.* (Left to right): *the an artist produced 2D patterns of darts, the corresponding 3D paper simulation, and the manufactured papercraft objects. We have yet to extend SC to incorporate the constitutive model of materials such as paper.*

Chapter 6

Guided Exploration of Physically Valid Shapes

This chapter describes techniques that generate information useful for creating designs from simulations, to facilitate intuitive exploration of physically valid shapes. We introduce algorithms that quickly produce suggestions and annotations by investigating physical constraints on the force domain, instead of the geometric domain. Many physical constraints are formulated as relationships among forces. For example, the condition for the yielding of material is given by the amount of stress. Force-space analysis investigates how force configurations change in accordance with changes in the design, using first-order approximations. We utilize force-space analysis to display parameter ranges as annotations during the design-editing phase, and generate suggestions for restoring structural soundness. Force-space analysis quickly provides approximations for stable designs, thus allowing annotations and suggestions to be presented to users in real time.

We investigate this technique in the context of nail-jointed furniture design that is especially aimed at producing unusual and artistic shapes with non-standard inclinations, while still ensuring physical validity. We impose two physical constraints: stability (the furniture stands without falling over) and durability (each nail can support the given weight without collapsing). We demonstrate that users can intuitively design stable and durable furniture utilizing these suggestions and annotations.

6.1 Introduction

Chapter 4 and Chapter 5 propose interactive frameworks to provide real-time feedback on physical constraints for various simulations. However, such methods only indicate whether or not the designed shape is valid; they do not suggest *how* to restore the model's validity. In a notable effort, Whiting et al. [172] directly optimized the procedure for generated buildings over a range of free variables to produce a final model that was structurally stable. However, such an approach is unsatisfactory for exploratory modeling, since it neither provides creative support, nor facilitates informed exploration. Our goal is to support *computational design* via real-time exploratory modeling [153, 163, 175], in which the system provides guidance to facilitate the user's manual design.

Given an initial shape and domain-specific geometric and physical constraints, we propose a computational design framework for efficient and intuitive exploration of valid shapes. Specifically, we actively guide the user in exploring those parts of the shape-space that satisfy the constraints, thus relieving him/her of the burden of ensuring realizability. We accomplish this via the following features: (i) We analyze the current shape configuration and display the valid range of the parameter being edited as an annotation. (ii) We also offer both continuous and discrete suggestions in coordinated editing modes, to restore validity when the current design is invalid. Note that in contrast to direct optimization-based solutions, we leave the designer in control of form-finding. We provide visualization of the valid range and multiple deformation suggestions, guiding the designer toward feasible geometric forms, as needed (see Figure 6.1).

In this work, we facilitate constrained modeling in the context of a (nail-jointed) furniture design system under geometric and physical constraints. Specifically, we consider three characteristics: (a) *connectivity* (the joint connections between planks are geometrically maintained), (b) *durability* (the object does not collapse at the joints under the target load distributions), and (c) *stability* (the object does not fall over or lose contact with the ground). The user interactively designs a shape model, utilizing standard modeling operations. In the meantime, continuous simulations of rigid bodies with frictional contact are continuously running in the background to provide real-time feedback on the structural validity of the design. The system performs sensitivity analyses to understand how design changes affect the *validity* of the design. We use this information to provide valid ranges for the parameters being edited, and also continuous suggestions for restoring validity via our novel force-space analysis. Each suggestion is offered in a coordinated editing mode that synchronously adjusts multiple components, which is otherwise a difficult task for the user, especially with nonlinear constraints. Thus, the user can efficiently navigate the physically valid shape-space by following the visualized ranges and exploring the suggestions (see Figure 6.19).

We employed our framework to design a range of furniture under different loads. Figure 6.2-(c) shows one of the pieces of furniture designed using our system. Our system supports real-time handling of up to 10 to 20 rigid bodies on a 2.7-GHz laptop. With our system, users can quickly and consistently design valid furniture, often with planks arranged in non-standard configurations. We fabricated a physical prototype and stress-tested it under the target specifications. We anticipate that our technique can easily be integrated with a range of modeling tools, enabling novel function-aware form-finding possibilities.

For example, IKEA [81] provides a range of specialized design-at-home tools

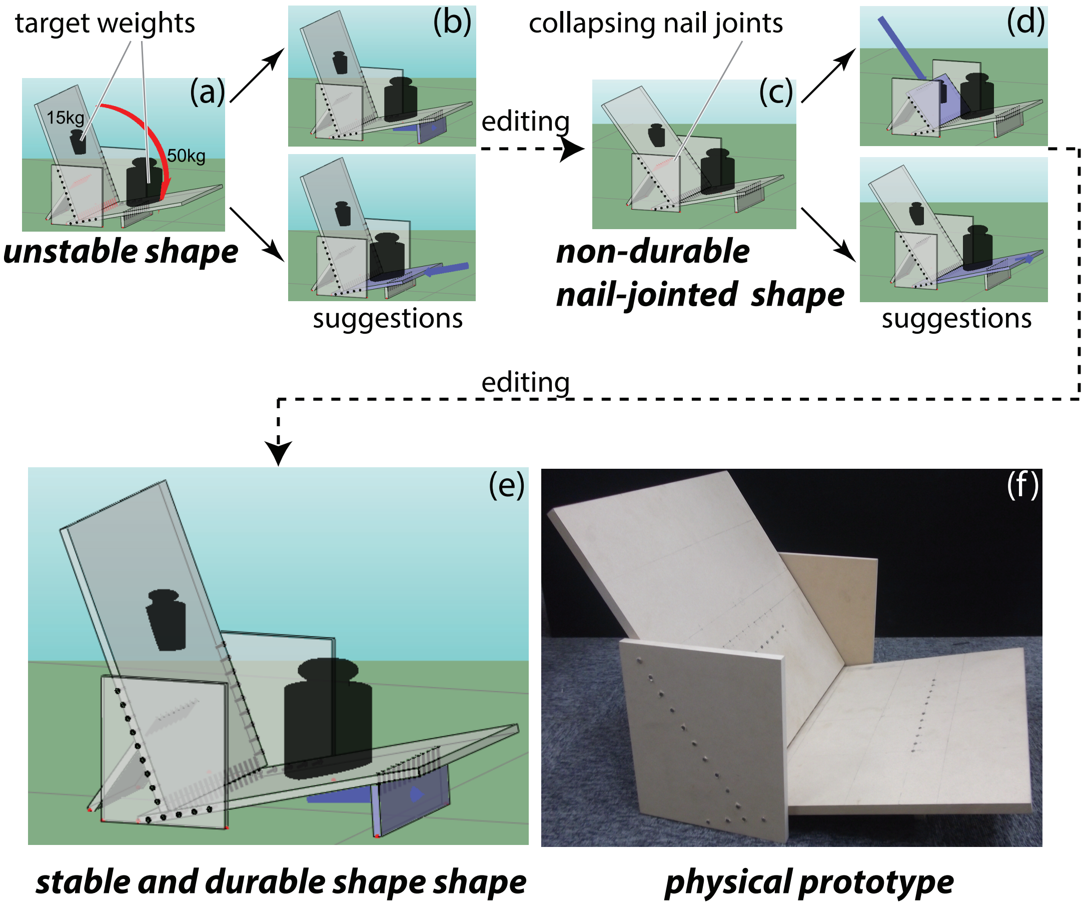


Figure 6.1: *Starting from a design (a) that is physically invalid due to model instability (i.e., falling over) or non-durability (i.e., excessive joint force), we offer design suggestions (b, d) for restoring physical validity. The suggestions provide guided shape-space exploration to the user, enabling him/her to quickly realize valid nail-jointed furniture designs in accordance with the weight-bearing targets and practical material specifications (e, f).*

for offices, kitchens, and bedrooms, allowing the user to prescribe room dimensions, interactively select 3D models from a product catalog, place them in a room, and plan a layout. However, such systems only permit users to choose from a list of fixed objects. With the growing demand for customization, an ideal system should also allow users to change the shape of the furniture, while still being guaranteed that the objects remain functional (e.g., bookshelves do not collapse under the target loads).

Contributions. In summary, we propose:

- an interactive modeling framework for designing valid shapes under geometric and physical constraints;
- a design environment for nail-jointed, plank-based furniture modeling with frictional contact, and implicit simulation of rigid-body motion; and

Figure 6.2: *An example of the furniture design process. (a) Furniture design begins with a sketch. (b) Using typical geometric modeling software may result in physically invalid furniture. (c) By using our system, a physically valid piece of furniture, similar to the original sketch, can be designed.*

- force-space sensitivity analysis to generate design suggestions with continuous and discrete modifications for restoring geometric and physical validity.

6.2 System Overview

Overview. Figure 6.3-left shows our modeling interface, which consists of a modeling panel and a suggestion panel. The modeling panel basically works as a standard modeling system (e.g., Google SketchUp), although it is specialized for models consisting of multiple planks connected by nail joints. Our system continuously checks for validity in the background and shows whether or not the configuration satisfies geometric and physical requirements. Specifically, the system examines connectivity, durability, and stability. Note that we do not check for self-intersections at runtime. The result of the analysis appears as an annotation in the main panel during mouse dragging. Further, we provide suggestions in the suggestion panel after mouse release if the current shape is invalid. Suggestions, when selected, appear in the modeling panel.

Modeling user interface. Figure 6.3-right shows the basic modeling operations supported by our system. The user draws two 2D lines on the screen to specify a new rectangular plank (a-c) of predefined thickness (12 mm in our setting). The first line is drawn by mouse dragging and is placed on a selected plank. The end point of

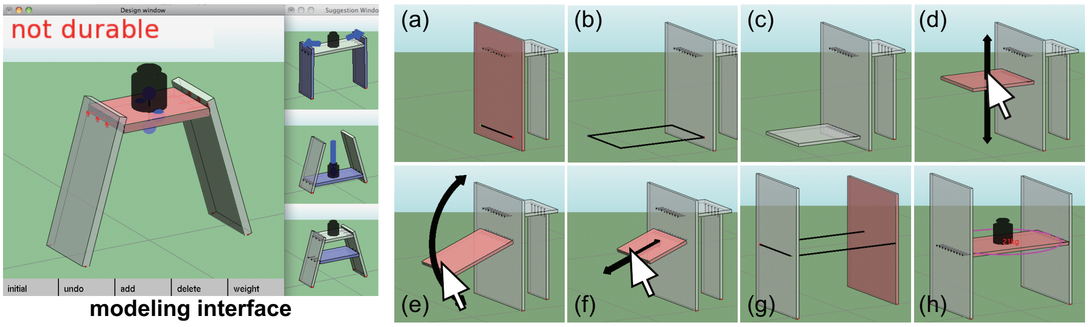


Figure 6.3: *(Left) The modeling interface consists of the modeling and the suggestion panels. (Right) The modeling interface with typical stages shown: creation, connection, translation, scaling, and rotation of a plank, along with placing a weight.*

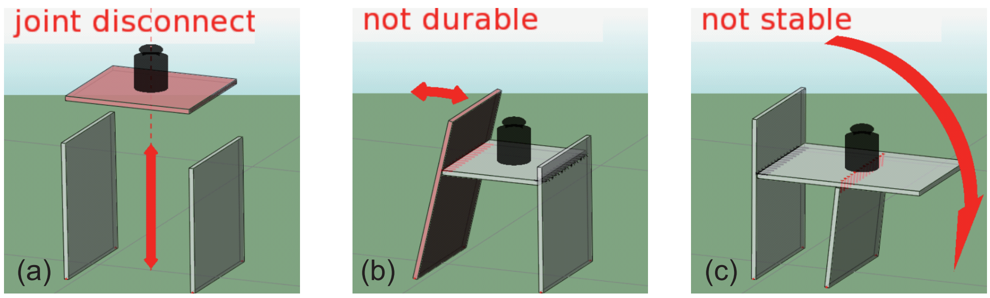


Figure 6.4: *Warnings flagged for invalid configurations: Joints get disconnected (a), a model becomes non-durable due to excessive force on the nails (b), or it becomes unstable, i.e., topples (c).*

the first line becomes the starting point of the second line and its end point is indicated by a mouse click. The second line is either projected to an existing plank or aligned to the canonical xyz-axis. We automatically generate a joint between the newly created plank and the existing planks on which the first and second line are placed. The user can translate, rotate, and scale a plank using 3D widgets (d-f). When an edge of a plank is placed near another plank, these planks are automatically connected (g). Finally, the user places a weight by clicking on a plank in the weight mode (h). Note that the exact placement of the weight on the selected plank is not important.

Validity visualization and suggestions. In Figure 6.4, we show the different scenarios when the current configuration becomes invalid. (a) When a joint becomes disconnected, the system highlights the joint in red. (b) When the model breaks at a joint, the system also highlights the joint in red. (c) When the model falls down, the system shows a red arrow. These warnings automatically appear and are continuously updated as the user interacts with the design, so that the user can move back to a valid state by direct manipulation based on the feedback.

In addition to checking whether or not the current configuration is valid, the sys-

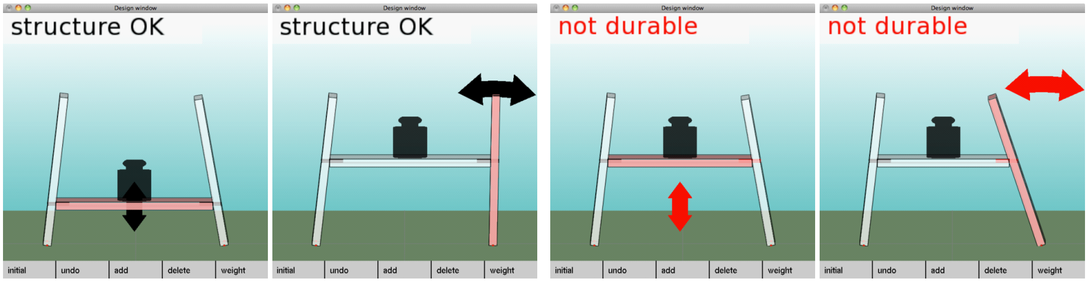


Figure 6.5: *Range indicators. Range is shown in black when the current configuration is valid and in red when invalid.*

tem computes the valid *range* of the parameter (degrees of freedom, DOF) being manipulated and shows it to the user during direct manipulation (mouse drag). When the current configuration is valid, the system shows the valid range in black. When the current configuration is invalid, the system shows the valid range in red (see Figure 6.5). Explicitly showing the valid range reduces the need for trial and errors to stay within or return to a valid state during direct manipulation editing.

When necessary, after each mouse release, the system provides suggestions (capped to a maximum of 8 in our setting) on how to resolve an invalid state. When a joint becomes disconnected, the system shows how to reconnect it (Figure 6.7a). When the model is non-durable or unstable, the system shows how to make it durable and stable (Figure 6.7b, 6.7c). Each suggestion consists of a representative configuration and an optional coordinated edit mode. When the user clicks on a suggestion, the representative configuration appears in the modeling panel together with arrow marks indicating the coordinated edits (Figure 6.6a). The user drags one of these arrows to make coordinated editing, thus allowing the user to control multiple DOFs of a model simultaneously while satisfying the required constraints. These multiple DOFs are coupled together, i.e., the user cannot fix the non-durability or instability moving each DOF individually. For example, in Figure 6.6, if the user slides the top board of the table toward the left, the angle of the left leg becomes perpendicular to the ground to compensate for the increase of the bending force on the left joint (Figures 6.6b, 6.6c).

6.3 Algorithm Overview

As the user edits the model (i.e., adds, removes, translates, rotates, or scales a plank), we first try to the satisfy geometric constraints, i.e., joint connectivity and ground contact, by adjusting the length of the other planks. If we fail to satisfy the geometric constraints, we suggest discrete changes to fix the design. After the model satisfies the geometric constraints, we check the physical validity of the current shape and present the result to the user. We test for durability and stability, which amounts

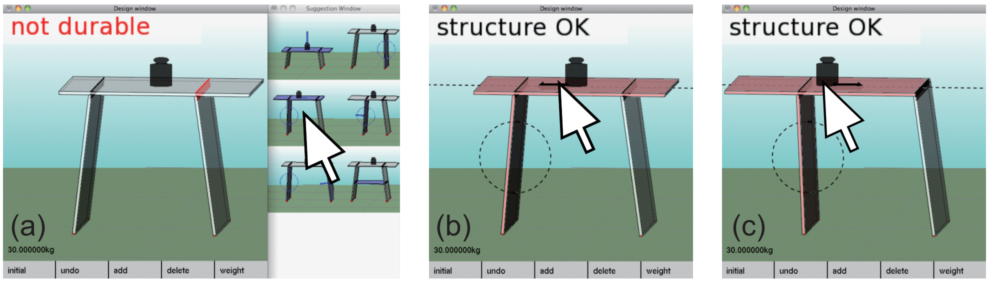


Figure 6.6: *Example of coordinated editing using suggestions. The table is non-durable and the system gives multiple suggestions (a). The user clicks on a suggestion and it appears in the modeling window (b). The user can change the position of the top board and left leg simultaneously by dragging any of the arrow handles (c).*

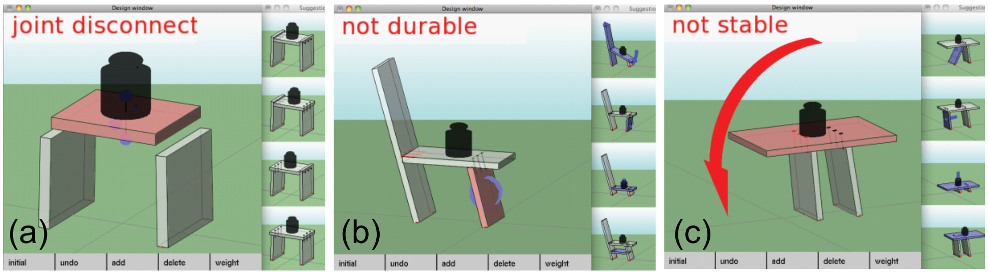


Figure 6.7: *Example of suggestions. A joint is connected (a), the model is made durable (b), and the model is made stable (c).*

to checking for inequality constraints on joint and contact forces, respectively. In addition to indicating that the design is valid or not, we also analyze how the validity changes with respect to further geometric modifications, i.e., what changes make the invalid model valid, and vice versa. The result of the analysis is used to compute valid ranges and make suggestions. Section 6.4 describes how we measure and analyze the physical validity, while Section 6.5 describes how we compute the valid range and make suggestions based on the analysis. Note that frictional contacts with the ground pose a challenge to the sensitivity analysis, and we present a method to address this issue.

6.4 Physical Validity

In our interactive framework, we continuously analyze the current design to provide feedback to the user about the physical validity of the current shape during the user's editing. Specifically, the system checks two types of physical validity: (i) whether or not the nail joint is durable, and (ii) whether or not the structure is stable. In this section, we first describe how to *measure durability* of a current design by solving constrained rigid body dynamics to obtain forces on the joint. Next, we propose a *sensitivity analysis* to analytically estimate changes in static equilibrium under infinitesimal perturbations of the current design. This analysis helps to generate editing suggestions as well as accelerate the computation of the validity.

6.4.1 Durability Measurement

In any nail-jointed wooden structure, the joints form the weakest links, i.e., such structures primarily break at the joints rather than at other sections [134]. Hence, in our framework, we model component planks of wooden furniture as assemblies of unbreakable rigid bodies, while focusing on the joint and the contact forces. We first define joint forces and then explain how to compute joint and contact forces for a given model. Next, we describe how to examine durability based on the obtained joint forces. Although most of the techniques explained in this section are standard in physical simulation, we describe them for completeness. An exception is the treatment of frictional contact. It is not trivial to handle frictional contact within the framework of sensitivity analysis and we present a novel method.

Notation. In the simulation of the nail-joined furniture, we consider two configurations of the rigid planks. The one is initial configuration and the other is deformed configuration. The initial configuration of the furniture is where the user designs a furniture and there is no gravitational force. On the other hand, the deformed configuration is where the gravitational force is applied and simulation achieves equilibrium status which gives deformation of nail joints and sagging of the floor. Here we described the notation of a plank's initial configuration and deformed configuration. Let us assume each rigid plank is labeled as P_i. There are three types of variables to represent the configuration of each rigid body; position, orientation, and size. Gravity center and orientation change from initial configuration to deformed configuration while size is constant throughout the simulation. The Figure 6.8 illustrate the notation of the configurations of a plank. In the initial configuration, gravity center of rigid body labeled as i is denoted as $\mathbf{c}_i \in \mathbb{R}^3$. Each planks have local coordinate. The initial orientation of rigid body i from spatial coordinate is denoted as $\mathbf{Q}^i \in SO(3)$. The orientation from the initial configuration to deformed configuration can be written as $\mathbf{R}^i \in SO(3)$. Hence, deformed plank has rotation from spatial coordinate as $\mathbf{R}^i\mathbf{Q}^i$. The planks are modeled as a cuboid aligned with rigid body's local axis. Each plank i has dimension $L_x^i \times L_y^i$ and with thickness of L_z^i which is simply written as three dimensional vector $\mathbf{L}^i = (L_x^i, L_y^i, L_z^i)$.

Definition of joint constraints. We characterize each nail-joint connection as a constraint between the participating plank pairs. We describe static rigid body equilibrium under joint constraints using standard notation (see Figure 6.9 and [66]). Let planks P_i and P_j be connected by a nail joint N_{ij}. Note that, although plank pairs are connected using several nails at a nail-joint, for simplicity, we represent such nail po-

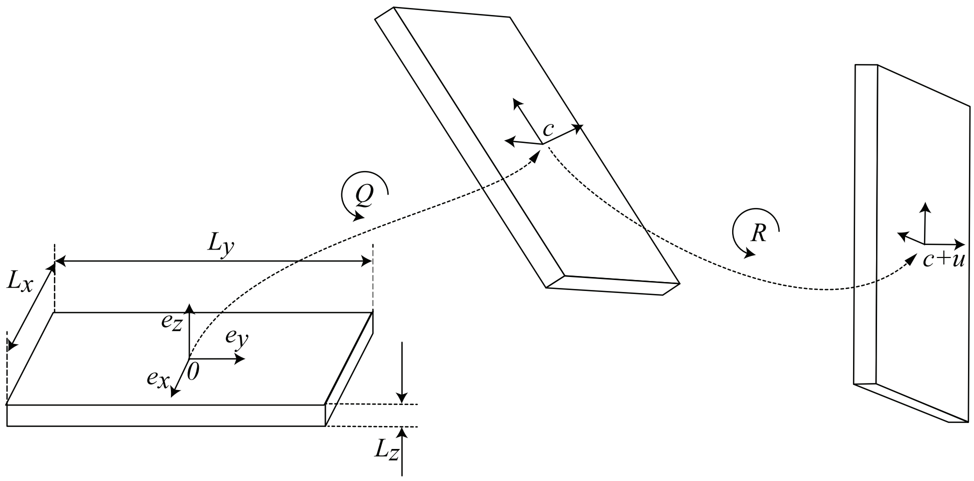


Figure 6.8: *The notation of the configurations of a plank.*

sitions using a single representative point $\mathbf{p}_{ij}$. The corresponding joint constraints are: (i) a translational part that keeps the participating planks together and (ii) a rotational part that prevents bending.

The translational part can be interpreted as keeping the joint representative point $\mathbf{p}_{ij}$ connected in the deformed configuration. This requires the deformed position of the joint representative point $\mathbf{p}_{ij}$ in the plank P_i and the plank P_j is identical:

$$\mathbf{d}_{ij}^{t} = [\mathbf{R}_i(\mathbf{p}_{ij} - \mathbf{c}_i) + \mathbf{c}_i + \mathbf{u}_i] - [\mathbf{R}_j(\mathbf{p}_{ij} - \mathbf{c}_j) + \mathbf{c}_j + \mathbf{u}_j]\,. \tag{6.1}$$

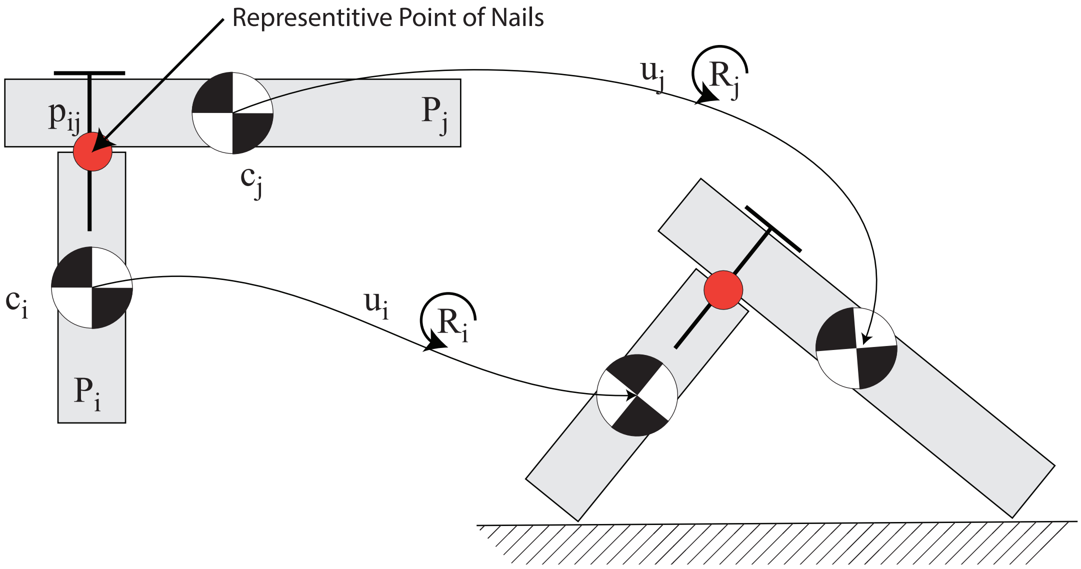


Figure 6.9: *The nail jointed structure in the initial configuration (left) and in the deformed configuration (right).*

We put bend-less constraint in the joint. We assume orthonormal coordinate $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ at joint in the initial configuration. The orthonormal coordinate undergo rotation in planks i and j as $(\mathbf{R}^i\mathbf{e}_x, \mathbf{R}^i\mathbf{e}_y, \mathbf{R}^i\mathbf{e}_z)$ and $(\mathbf{R}^j\mathbf{e}_x, \mathbf{R}^j\mathbf{e}_y, \mathbf{R}^j\mathbf{e}_z)$ and these two deformed coordinates should be equal. Hence the bend-less constraint holds $\mathbf{\Phi}_b = 0$ where,

$$\mathbf{d}^r_{ij} = \begin{pmatrix} (\mathbf{R}^i\mathbf{e}_y)^T(\mathbf{R}^j\mathbf{e}_z) \\ (\mathbf{R}^i\mathbf{e}_z)^T(\mathbf{R}^j\mathbf{e}_x) \\ (\mathbf{R}^i\mathbf{e}_x)^T(\mathbf{R}^j\mathbf{e}_y) \end{pmatrix}. \tag{6.2}$$

Here, the value of $\mathbf{d}^t_{ij}$ amounts to infinitesimal rotation vector which denote how much the joint is bend around the spatial axis $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$. The Equation 6.2 can be more compactly written as

$$\mathbf{d}^r_{ij} = vect\left(\mathbf{R}_i^T\mathbf{R}_j\right), \tag{6.3}$$

where $vect$ is an operator that extracts the axial rotation vector of a rotation matrix. Note that since both $\mathbf{R}_i, \mathbf{R}_j \in SO(3)$ are rotation matrices, $\mathbf{R}_i^T\mathbf{R}_j$ is also a rotation matrix. At each nail-joint N_{ij} the joint constraints are:

$$\mathbf{d}^t_{ij} = 0 \quad \text{and} \quad \mathbf{d}^r_{ij} = 0. \tag{6.4}$$

The set of such constraints for a piece of furniture can be redundant (e.g., if a set of planks is connected in a loop) leading to an over-constrained system. As a solution, we allow for deviations from the exact constraints using a penalty method. Specifically, we measure deformation energy at joint N_{ij} as

$$E^{joint}(N_{ij}) = 0.5\|\mathbf{d}^t_{ij}\|^2/\epsilon^t + 0.5\|\mathbf{d}^r_{ij}\|^2/\epsilon^r, \tag{6.5}$$

which we include as the potential energy of the system (see Equation 6.11). The scalar values ϵ^t and ϵ^r are small constants (both set to 10^{-5} in our tests). The derivative of penalty function E^{joint} with respect to $\mathbf{d}^t$ and $\mathbf{d}^r$ are

$$\mathbf{h}^t = \mathbf{d}^t/\epsilon^t \quad \text{and} \quad \mathbf{h}^r = \mathbf{d}^r/\epsilon^r, \tag{6.6}$$

and can be seen as constraint forces. We call such forces translation forces and bending forces (in engineering, commonly referred to as the bending moment), respectively. Note that these deviations $\mathbf{d}^t$ and $\mathbf{d}^r$ are influenced by the values of ϵ^t and ϵ^r, but $\mathbf{h}^t$ and $\mathbf{h}^r$ are not. The $\mathbf{h}^t$ and $\mathbf{h}^r$ have physical meaning relating to the equilibrium of the forces between planks.

Contact and friction. Coulomb's friction model with static kinetic parameter μ_s is applied for our physics model. There are a number of frictional contact models but most of them is for dynamic motion simulation. In our application, we are interested in static behavior of furniture rather than dynamic animation. Hence, we modeled both contact and friction constraints using penalty method, where friction forces are determined by static problem assuming applying frictional springs. We consider that the floor is flat and thus, can be represented with a normal $\mathbf{n}$ and a point on the floor $\mathbf{x}_{floor}$. Since the floor is flat, we can only care the convex hull of the plank to avoid penetration. Hence, we model that frictional contacts occur at plank's four corner points written as a in the plank's local coordinate. The contact force can be written as

$$\mathbf{f}_n = k_n h_{penetration} \mathbf{n}, \tag{6.7}$$

where the $h_{penetration}$ is a penetration depth given by $h_{penetration} = \mathbf{n}^T \left(\mathbf{x}_{floor} - \mathbf{x}_{contact}\right)$ and $\mathbf{x}_{contact}$ is a contacting point on the plank given as $\mathbf{x}_{contact} = \mathbf{R}^i \mathbf{Q}^i \mathbf{a} + \mathbf{c}^i + \mathbf{u}^i$. This contact force can seen as a Jacobian of a contact energy:

$$E^{penetration} = \frac{1}{2} k_n h^2_{penetration}. \tag{6.8}$$

Similarly, the frictional force is generated from a friction spring that is placed tangent to the floor:

$$\mathbf{f}_s = -k_f (\mathbf{I} - \mathbf{n} \otimes \mathbf{n})(\mathbf{x}_{contact} - \mathbf{x}_{anchor}), \tag{6.9}$$

where $\mathbf{x}_{anchor}$ is a point on the floor where the contact point touch the floor for the first time. This static friction force can be seen as a Jacobian of the friction energy:

$$E^{friction} = \frac{1}{2} k_f \left\| (\mathbf{I} - \mathbf{n} \otimes \mathbf{n})(\mathbf{x}_{contact} - \mathbf{x}_{anchor}) \right\|^2. \tag{6.10}$$

We consider the total frictional contact energy amounts to be $E^{contact} = E^{penetration} + E^{friction}$.

Computation of Joint and Contact Forces. In this work, we focus on the behavior of shapes under static equilibrium rather than dynamic motion of rigid bodies. We therefore compute forces applied to each joint by directly minimizing the total potential energy of the system with respect to $\mathbf{u}$, $\mathbf{R}$, and $\mathbf{h}$:

$$E^{total}(\mathbf{u}, \mathbf{R}, \mathbf{h}) = -\sum_{i}^{|P_i|} M_i \mathbf{c}_i^T \mathbf{g} + \sum_{ij}^{|N_{ij}|} E^{joint}(N_{ij}) + \sum_{k}^{|N_{contact}|} E_k^{contact}, \tag{6.11}$$

where M_i is the mass of plank P_i and $\mathbf{g}$ is acceleration due to gravity. The first term captures the gravitational potential energy; the second term models the joint energy; while the last term is due to contact forces as described later (we weigh the terms equally). With the total potential energy E^{total} being nonlinear, we iteratively minimize the potential energy using the Newton-Raphson method. Since the Hessian of the penalty term E^{joint} is ill-conditioned, we treat the constraint forces $\mathbf{h}^r$ and $\mathbf{h}^t$ as independent variables and explicitly solve for them. Specifically, we simultaneously minimize with respect to $\mathbf{h}^t$ and $\mathbf{h}^r$ along with $\mathbf{u}_i$ and $\mathbf{R}_i$ (see Equation 6.6). Note that since the Hessian of the total potential energy is indefinite, we damp the iteration by adding an identity matrix scaled by a small value (we use 10^{-5}) to the diagonal component of the translation and rotation for each plank. We continue the iteration until variables $\mathbf{u}$ and $\mathbf{R}$ stabilize. The solution to this minimization problem yields deformed plank positions $\mathbf{u}$, $\mathbf{R}$, and the joint force $\mathbf{h}$.

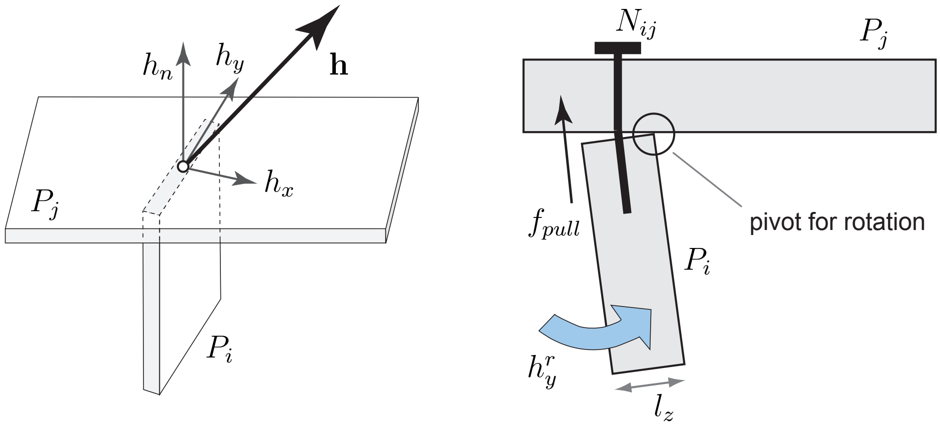


Figure 6.10: *(Left) Decomposition of a constraint force into components in local coordinates. (Right) The rotation force causes the nail pulling force, which can affect the joint's durability.*

Joint durability. Having the translation $\mathbf{h}^t$ and the bending force $\mathbf{h}^r$ at each joint, we check for the durability of the nail-joint under the given forces. Mechanical properties of nails are well understood and have long been standardized with precise specifications on their load-bearing capacities (see [22]). At any joint, the loads on the nails are of two types: (i) a pulling force, which acts along the axis of the nail, and (ii) a shearing force, which acts vertical to the nail axis. We express force $\mathbf{h}$ (i.e., $\mathbf{h}^t$ and $\mathbf{h}^r$) in a local coordinate system: h_n represents the component normal to the joint face for plank P_j, h_x the component in a direction along the normal of P_i, and the remaining component is h_y (see Figure 6.10). Each component of $\mathbf{h}^r$ denotes the torque to twist the plank P_j with an axis of rotation in each direction. We assume the plank's thickness is smaller than the width of the joint between P_i and P_j. Hence, the bending force h_y dictates the collapse of the joint. In Figure 6.10, we show how the nail-pulling force arising from bending force h_y^r is modeled. The joint forms a lever with the length of the lever arm equal to $0.5l_z$. Specifically, we model the pulling force as

$$f_{pull} = \frac{1}{N_{nail}} \left(2|h_y^r|/l_z - h_n^t \right), \tag{6.12}$$

where l_z represents the thickness of the plank (12 mm in our tests) and N_{nail} denotes the number of nails at the nail joint N_{ij}. Then, the shear force is given by

$$f_{shear} = \frac{1}{N_{nail}} \sqrt{{h_x^t}^2 + {h_y^t}^2}. \tag{6.13}$$

Finally, we mark a joint as durable if both forces are within allowable threshold margins [22].

6.4.2 Sensitivity Analysis

We now investigate how design changes affect the physical validity of a shape as this helps to accelerate the force computations (solving Equation 6.11) as the user changes the design. More importantly, when needed, the sensitivity analysis helps in generating suggestions for changing the design to restore validity by making the model durable and stable. Specifically, we locally compute a linear approximation to study how forces in equilibrium change with respect to changes to the current design, i.e., we perform a sensitivity analysis [165].

Let γ represents a shape configuration (see also Section 6.5). Using implicit rigid body analysis, the static equilibrium can be expressed as a linear system: $\mathbf{A}(\gamma)\mathbf{x}(\gamma) = \mathbf{b}(\gamma)$, where $\mathbf{A}$ is a square matrix and $\mathbf{x}$ is a vector encoding the positions and orientations of all the planks along with the forces $\mathbf{h}$ at the different joints. Vector $\mathbf{b}$ stores the generalized external forces, i.e., forces due to gravity, contact, and friction acting on the planks. The sensitivity analysis gives

$$\frac{d\mathbf{x}}{d\gamma} = \mathbf{A}^{-1}\frac{d\mathbf{b}}{d\gamma}. \tag{6.14}$$

Although the configuration $\mathbf{x}$ changes nonlinearly with respect to any initial design change $\delta\gamma$, we found it sufficient to use $\mathbf{x} \rightarrow \mathbf{x} + d\mathbf{x}/d\gamma \cdot \delta\gamma$ as an initial guess to bootstrap the nonlinear iteration and achieve faster convergence. Detail of the sensitivity analysis under an hard constraint is described in the Appendix C.

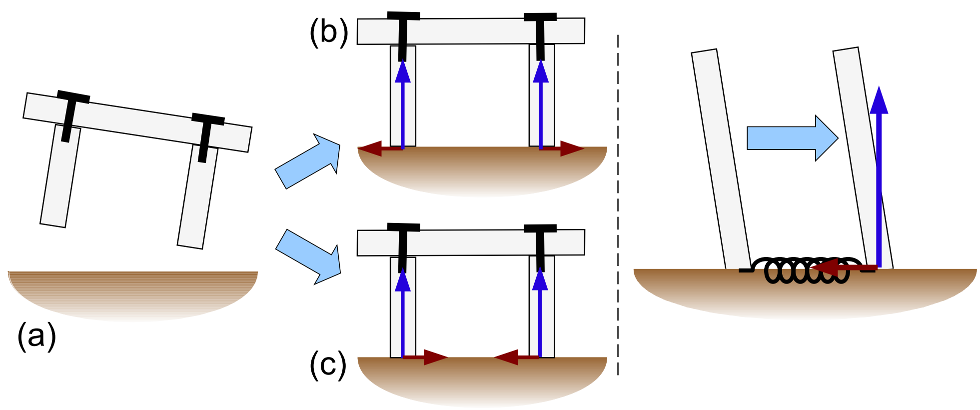


Figure 6.11: *(Left) Redundancy of the frictional contact forces. The red arrows show the friction forces and the blue arrows show the contact forces. A table with initial configuration (a) falls to the ground and can have multiple possible friction forces like (b) or (c). (Right) The penalty-based frictional force determines a unique friction force as a deviation from the initial position.*

Sensitivity analysis of frictional contacts. We assume that the design structure is casually placed on the ground and not bolted to it. Hence, friction is essential to prevent sliding under horizontal force. For example, a table depends on friction to resist sliding under horizontal forces, say when we push the table sideways. Although a table with vertical planks as legs can easily support vertical loads, it is fragile even under slight horizontal perturbation, which is undesirable.

Performing an accurate sensitivity analysis with frictional contacts is challenging because frictional forces depend on the direction of the tangent velocities at the contact points. Sensitivity analysis, however, assumes static equilibrium with zero velocity at the contact points and hence cannot be used to determine friction force directions. Further, redundancy among frictional forces poses additional challenges [92], e.g., even if a chair stands still, the combination of frictional forces is unknown, making it difficult to determine the internal forces (see Figure 6.11-left).

We propose a simple penalty-based method to address the above problems. In a standard dynamic setting, friction anchors are placed at the impact location and are relocated as the contact points slide with kinetic friction [54]. However, since our setting is static, we assume that (i) all contact points are exactly on the ground and (ii) the contact states do not change during interactions. This allows us to uniquely determine the anchor position with respect to the initial configuration (see Figure 6.11-right) and analyze frictional force under design changes. Specifically, we place the contact points at the corners of planks that touch the ground. When the user sketches a plank, we detect the plank corner that touches the ground, and mark it as a contact. Note that during design changes we ensure that the contacts touch the ground without penetration or floating in the air (see Section 6.5). For sliding, we relocate friction anchors so that the (friction) springs do not generate excessive force beyond the limit of Coulomb friction.

6.5 Exploration of Valid Spaces

In this section, we describe how our framework guides the user towards the *valid* subspace of the configuration space Γ. If the current design is valid, we indicate the range of user manipulations that keeps the design validity. On the other hand, when the current design becomes invalid, we make multiple suggestions to restore validity. Note that even though the (unconstrained) configuration space is high-dimensional, our computational framework only exposes meaningful (i.e., valid) suggestions, thus greatly simplifying the user's task. We make both continuous and discrete suggestions: while continuous suggestions leave the inter-plank joint topology unchanged, discrete suggestions involve adding support materials.

6.5.1 Geometric Constraints

Aside from the physical validity of the shape, i.e., its durability and stability, shapes designed in our system are geometrically restricted by two constraints: (i) geometrical joint constraints and (ii) contact constraints (see Section 6.4). We first restrict the design space where the shape satisfies these geometrical constraints and then investigate the physical validity. Each plank has 8 degrees of design freedom: 3 for translation, 3 for rotation, and 2 for edge lengths around the plank faces (the plank thickness is fixed). For each degree of design freedom of the planks, we ensure that the contact constraints and joint constraints are satisfied by adjusting the length of the planks (Figure 6.13-left). Further, some degrees of freedom are invalid, e.g., if both sides of a plank are nailed, the plank length cannot be adjusted (Figure 6.13-right). We identify and remove such invalid degrees of freedom from the design space. Note that if there are C number of plank components and #DOF$_{invalid}$ number of invalid design degrees of freedom, the constrained design space Γ has dimensions of $N_\gamma = 8C-$#DOF$_{invalid}$. Each basis corresponds to one plank's translation, rotation, or length change and the adjacent planks' length change. We scale the translation and length change basis with the inverse of the size of the maximum bounding box edge length to make the translation and length change DOFs dimensionless, like that of rotational DOFs. Next, we enable exploration in a physically valid subspace of a constrained design space Γ.

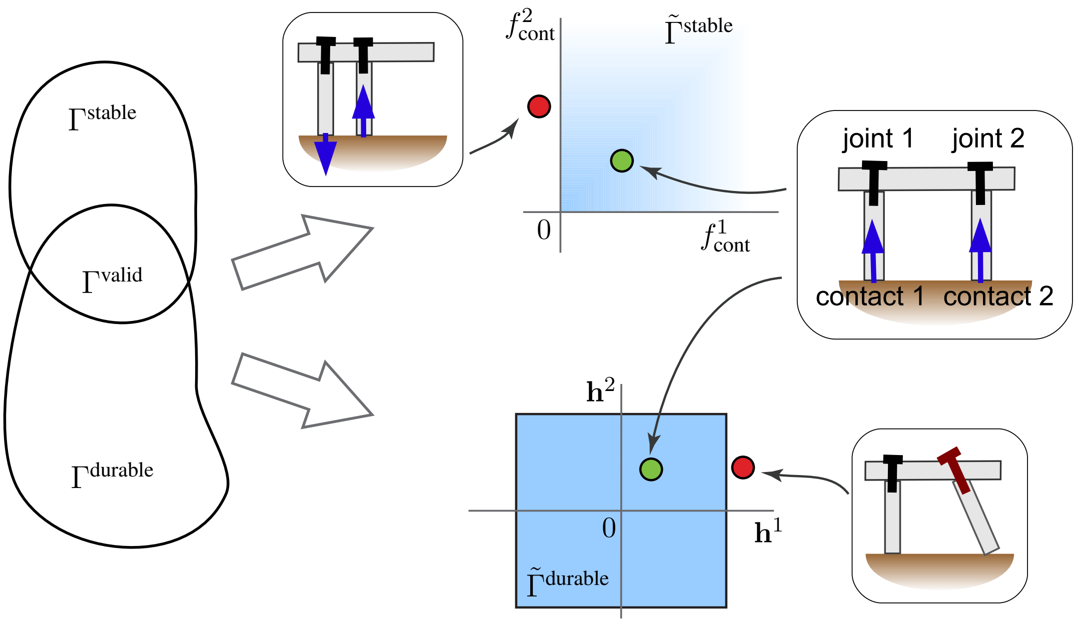


Figure 6.12: *A shape space point is valid if it is both stable and durable. For invalid shapes, we propose deformation suggestions to return to the valid part of the shape space. We work in force spaces defined by contact forces and bending forces for stability and durability, respectively thus simplifying the problem. Specifically, stability amounts to contact forces being restricted to the first quadrant, while durability amounts to bending forces being restricted to a durability rectangle. Note that although in this example the force spaces are 2D, in general we work in high-dimensional spaces.*

6.5.2 Valid Space

Recall that a shape is physically valid if two conditions are satisfied: (i) the shape is *durable*, which amounts to each joint having both pulling and shear forces below allowed thresholds, written as

$$|f_{pull}| \leq f_{pull\ max} \text{ and } |f_{shear}| \leq f_{shear\ max} \quad \forall N_{ij}, \tag{6.15}$$

and (ii) the shape is *stable* (i.e., it does not topple), which amounts to each contact point having a non-negative contact force f_{cont} in the direction normal to the ground, written as

$$f^l_{cont} \geq 0 \ \ \forall \text{ contact points } l. \tag{6.16}$$

Let the corresponding subspaces of the configuration space Γ be $\Gamma^{durable}$ and Γ^{stable}, respectively. Thus, the valid shape space is $\Gamma^{valid} = \Gamma^{durable} \cap \Gamma^{stable}$. When the current design becomes invalid, the goal is to provide multiple suggestions to return back to the valid shape space (see Figure 6.12).

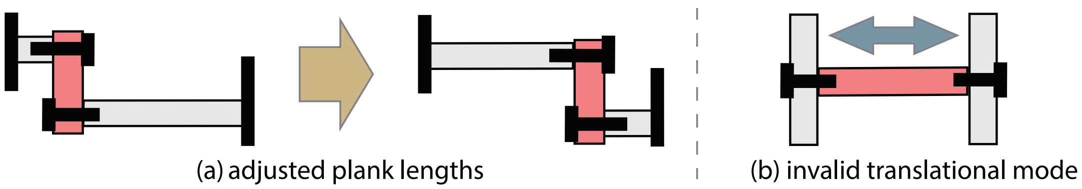


Figure 6.13: *Constrained design modes: (a) the lengths of neighboring planks of the edited planks are adjusted so that joints stay connected; (b) a translation mode is invalid if both sides of the planks are jointed.*

The valid space typically has a complex boundary since it is characterized by non-linear inequality constraints. Further, since the configuration space is high-dimensional, computing the exact boundary is difficult and time consuming. Also, it is nearly impossible to arbitrarily pick a valid shape directly from the high-dimensional space Γ^{valid}. Instead, we first pick several meaningful search directions to pursue, i.e., directions such that the invalid shape becomes valid under small manipulations. For each such direction, we use line search to identify configuration intervals where all the validity conditions are satisfied.

Since the boundaries of $\Gamma^{durable}$ and Γ^{stable} are characterized by force inequalities, we consider the valid shape space boundary in the *force space*, i.e., a coordinate space with the forces as the axes. This simplifies the problem as the boundary is then geometrically prescribed by the corresponding inequality. For example, with two contact points, the stable region is 2D and Equation 6.16 simply indicates that the first quadrant is the stable region (see Figure 6.12).

To efficiently characterize the joint durability force space, we make two approximations: (i) the translation force $\mathbf{h}^t$ in Equation 6.12 remains constant with respect to small design changes and only the bending force $\mathbf{h}^r$ varies and (ii) the shearing force in Equation 6.13 does not change under small design changes. These approximations are true when the bending force $\mathbf{h}^r$ is dominant and more sensitive than $\mathbf{h}^t$ under design changes. Thus, Equation 6.15 becomes

$$|h_y^r| \leq 0.5 l_z \left(f_{pull\ max} N_{nail} + h_n^t \right) = \Lambda_{max}. \tag{6.17}$$

Geometrically, the stable region Γ^{stable} is approximated as a high-dimensional, axis-aligned cuboid with edge lengths of Λ_{max} and centered at the origin in the joint bending force space (see Figure 6.12).

Note that the dimensions of the contact force space and the bending force space are lower than the configuration space dimension ($|\Gamma| \approx 8C$). Specifically, the contact force space has a dimension of the number of contact points, while the joint bending space has a dimension of the number of joints N_{ij}. Next, we describe how to efficiently search for directions in this simplified representation. We denote the boundaries of the stable and durable force space as $\tilde{\Gamma}^{stable}$ and $\tilde{\Gamma}^{durable}$, respectively.

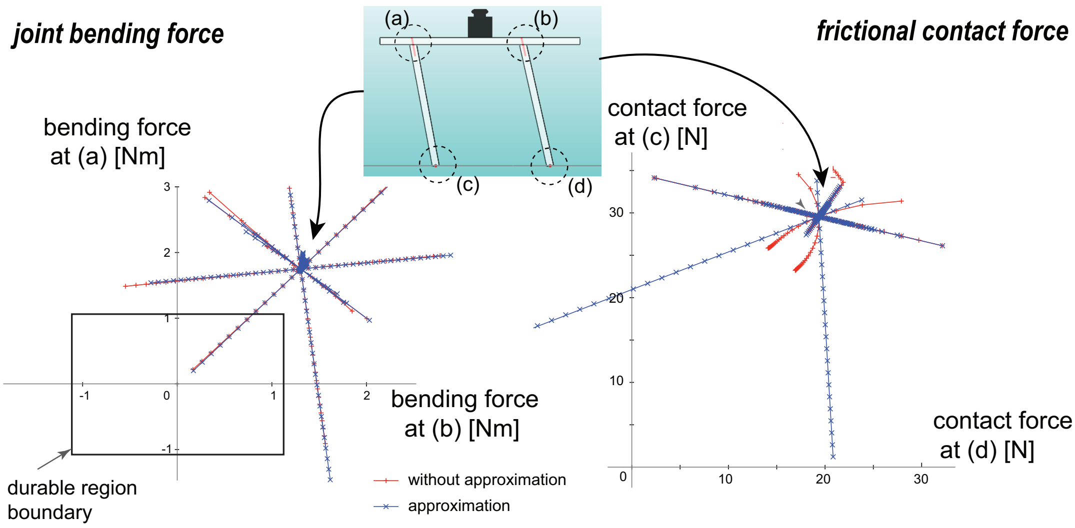


Figure 6.14: *Comparison of a stable shape and a durable space with and without approximation.*

6.5.3 Visualization of the Valid Range

During direct editing, we display the valid range of the parameter being manipulated. To do this, we evaluate the validity by changing the parameter. When the current configuration is already valid, the search proceeds in both directions until it becomes invalid to identify the bounds. When the current configuration is invalid, we first select the direction of the search using the result of the sensitivity analysis and then run a bisection search along that direction to identify the valid range, as explained next.

6.5.4 Continuous Shape Suggestions

When the current configuration becomes invalid, we compute several suggestions: (i) if only stability is violated, the system finds search directions to restore stability by analyzing the boundary of $\tilde{\Gamma}^{durable}$; (ii) if only durability is violated, the system finds search directions to restore durability by analyzing the boundary of $\tilde{\Gamma}^{stable}$; and (iii) if both stability and durability are violated, the system first proposes directions to restore stability and then restore durability.

Under local changes, we assume forces to vary linearly according to the design variations. Note that we use linearization only to select good directions (see Yang et al. [175] for use of higher-order derivatives). After selecting a direction, we run a bisection search along the direction using all the nonlinear constraints without any approximation for actually computing valid designs. Figure 6.14 shows a sample comparison with and without linear approximation. If we cannot find valid shapes in the direction, we simply omit it from the suggestions (see Algorithm 1).

Algorithm 1 Generating durability-restoring suggestions.

Generate design modes $\{\gamma_0, \ldots, \gamma_{N_\gamma}\}$
Compute $\mathbf{A}^{-1}$ {In Equation 6.14}
Generate sensitivity of $\mathbf{h}, \mathbf{f}_{contact}$ against all Dofs in Γ
C : set of combination of integer value
for $m = 1$ to M **do**
  for m number combination of modes $c = \{i_1, \ldots, i_m\}$ **do**
    if all subset of c is not in C **then**
      Compute $\mathbf{K}_0$ and $\mathbf{y}^\star$ {Equation 6.19}
      Compute $t^\star$ {Equation 6.18}
      if $t < 1$ **then**
        C ← C ∪ c
      end if
    end if
  end for
end for
For $c \in C$, find range of durable shapes using bisection method
Order the suggestions based on the computed range

We found that simultaneously exploring the full design space involves searching over $|\Gamma| \approx 8C$ dimensions, which is impractical for real-time performance. Also, users can find suggestions involving variations across many parts to be confusing. Instead, we focus on suggestions involving at most M degrees of freedom for any suggestion (3 in our examples). We try all the possible combinations of selecting m design DOF-s ($m \leq M$), denoted by $\{\gamma_1, \ldots, \gamma_m\}$. We parameterize a search direction as a unit vector $\mathrm{s} \in \mathbb{R}^m$ with coordinates s_i such that $\sum_{i=1}^{m} s_i^2 = 1$ and the direction is $\mathrm{s} = \sum_{i=1}^{m} s_i \gamma_i$.

Durability-restoring suggestions. A desirable search direction s should quickly make the design durable, i.e., reach the boundary $\tilde{\Gamma}^{durable}$. Thus, for any direction s we look for

$$t^{\star} = \arg\min_{t} t\mathbf{K}_0\mathbf{s} + \mathbf{h}_{y0}^r \in \tilde{\Gamma}^{durable}, \tag{6.18}$$

where matrix $\mathbf{K}_0 \in \mathbb{R}^{N_{ij} \times m}$ defines sensitivities of joint forces with respect to design changes $\mathbf{K}_0 = \nabla \mathbf{h}_y^r = \left[\partial \mathbf{h}_y^r / \partial \gamma_1, \ \ldots, \ \partial \mathbf{h}_y^r / \partial \gamma_m\right]$ evaluated at the current joint bending force $\mathbf{h}_{y0}^r$. For interactive performance, instead of finding the minimum step t along direction s, we compute the search direction s that takes us closest to the origin. Specifically, we choose a direction such that

$$\mathbf{y}^{\star} = \arg\min_{\mathbf{y}} \left\| \mathbf{K}_0\mathbf{y} + \mathbf{h}_{y0}^r \right\|, \quad \mathbf{y} \in \mathbb{R}^m, \tag{6.19}$$

and use $\mathrm{s} = \mathbf{y}^{\star}/\|\mathbf{y}^{\star}\|$ to compute t using Equation 6.18. We use a brute force method and try all the possible $M!$ combinations in the configuration space taking advantage of the simple axis-aligned cuboid approximation of the durable region. Specifically, finding the search direction can be seen as detecting collisions of rays with the durability cuboid where sensitivity of the bending force $\mathbf{K}_0\mathrm{s}$ acts as a ray with its source at $\mathbf{h}_{y0}^r$ (see the Figure 6.15). Hence, we cull a direction if either the norm of the sensitivity $\mathbf{K}_0\mathrm{s}$ is small (≤ 1 in our tests) or the ray faces away from the cuboid.

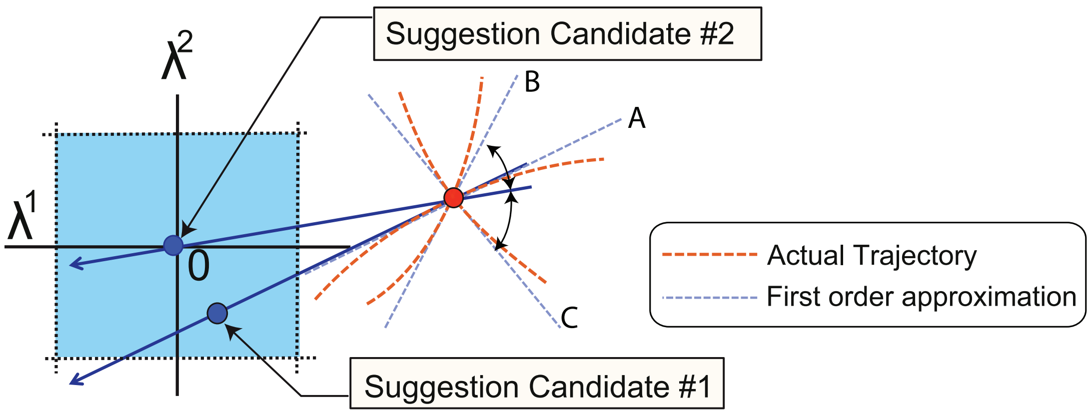


Figure 6.15: *Our suggestion generation algorithm for restoring durability in a simple structure with two nail joints. First, all the first order approximation of trajectories in the force space (i.e. ray) are computed, and then the system finds possible combination of the rays that reaches the durable region (shown in a blue rectangular). The ray A in the figure goes directory inside the durable region hence it is presented as a suggestion. The combination of rays B and C yet goes inside the durable region so it also presented as a suggestion.*

Stability-restoring suggestions. We compute stable shape suggestions similar to the durability case. Specifically,

$$t^{\star} = \arg\min_{t} t\mathbf{L}_0\mathbf{s} + \mathbf{f}_{cont0} \in \tilde{\Gamma}^{stable}, \tag{6.20}$$

where $\mathbf{L}_0 \in \mathbb{R}^{N_{contact} \times m}$ is a sensitivity matrix of contact forces with respect to design changes $\mathbf{L}_0 = \nabla \mathbf{f}_{cont} = [\partial \mathbf{f}_{cont}/\partial \gamma_1, \ \ldots, \ \partial \mathbf{f}_{cont}/\partial \gamma_m]$ evaluated at the position of the current contact force $\mathbf{f}_{cont0}$. We choose a direction $\mathbf{s}$ such that the shape quickly becomes stable. First, we project the current contact force vector $\mathbf{f}_{cont0}$ on the stable region to obtain $\mathbf{f}^{*}_{cont0}$ (i.e., clamped to zero) and choose the direction that gets us closest to $\mathbf{f}^{*}_{cont0}$ using a least squares minimization, i.e., $\mathbf{s} = \mathbf{y}^{\star}/\|\mathbf{y}^{\star}\|$ such that

$$\mathbf{y}^{\star} = \arg\min_{\mathbf{y}} \|\mathbf{L}_0\mathbf{y} + \mathbf{f}_{cont0} - \mathbf{f}^{*}_{cont0}\|, \quad \mathbf{y} \in \mathbb{R}^m. \tag{6.21}$$

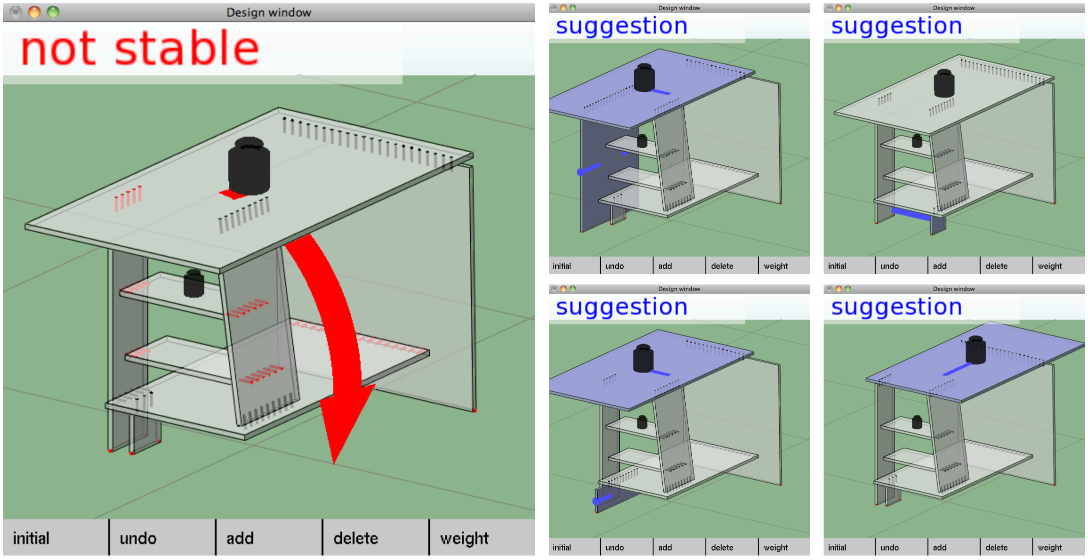


Figure 6.16: *Stability-restoring suggestions.*

6.5.5 Discrete Shape Suggestions

When the structure is not durable, we try to make it durable by adding a support plank as a reinforcement around a joint that is under excessive force. Typically, nail joints connect two planks nearly at a right angle, making it difficult to attach any support material between the planks connected by the non-durable joint. Instead, we try to connect two planks that are parallel to each other and put the supporting plank orthogonal to the planks. We use a greedy strategy. First, we choose a combination of two planks P_i, P_j such that (i) they are nearly parallel (we use $|\mathbf{n}_i \cdot \mathbf{n}_i| < 0.5$ where, $\mathbf{n}_i$ is the face normal of plank P_i, and the same for $\mathbf{n}_j$), (ii) between the two planks there is a third plank P_k connected to P_i and P_j by joints, and (iii) one or both joints

N_{ik} and N_{jk} are non-durable. We suggest adding support material between P_i and P_j at a location chosen from several (rule-based) candidate positions so that the support material does not intersect with other planks (see Figure 6.17). We check for durability of the joint by running a physical simulation to make sure that the support plank is effective. The system tries many combinations of planks until it finds effective supporting planks based on standard rules used in woodwork [22]. Smarter strategies should be investigated in the future.

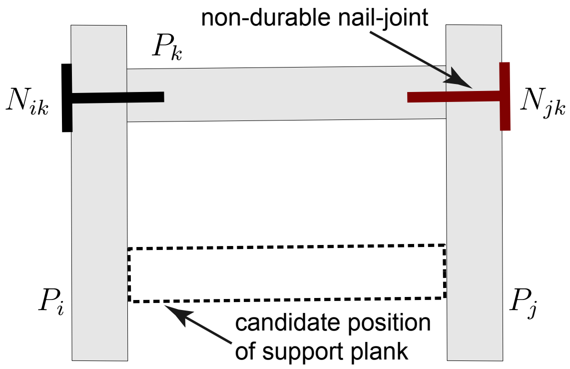


Figure 6.17: *Heuristic to add a support plank.*

6.6 Results

In our system we consider furniture designs using 12 mm medium density fiberboard (MDF) with 32 mm nails, spaced at interval of 20 mm. Such a placement can take a maximum shear force of $f_{shear\ max} = 190$N and maximum pull force of $f_{pull\ max} = 35$KN/m [134]. We set the coefficient of static friction to 0.5 in our tests. In our current implementation, we can regularly handle up to 10-15 plank designs at interactive speed. In each exploration session, the user progressively adds planks and proposes an initial configuration with the target load-bearing capacity. For example, in Figure 6.1, we put 50 kg weight on the horizontal plank and 15 kg on the supporting back plank. The final design was found after several iterations of suggestions and design explorations. We built a physical prototype (the construction took around 4 hours) and found it to behave satisfactorily under the target load.

In Figure 6.18, we use our system to design non-conventional bookshelves. The computational support is critical as we have little intuition in such unusual situations and cannot benefit from prior experience. Guided exploration helps the user to explore the design limits while not having to worry about physical validity.

Figure 6.19 shows additional design sessions with our system. Note that we show only a few representative suggestions. The user is provided with corrective suggestions *only* when the design becomes invalid. Further, each suggestion comes with a range where the shape remains valid. Thus, even when the suggestion modes involve

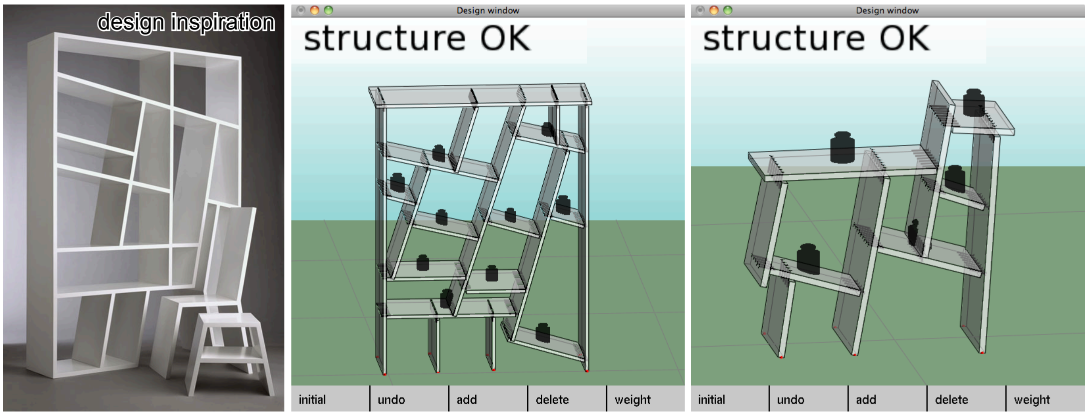


Figure 6.18: *(Left) Designing non-standard furniture is difficult for novice users. Our guided exploration framework allows users to design strange configurations easily with target load specifications (Middle and Right).*

multiple planks, the user simply has to adjust a *single* parameter along the suggested deformation direction. For example, in the chair 1, we show 3 different suggestions each involving a pair of planks to be simultaneously manipulated to restore validity. In case of chair 2, the situation is similar, but we have 3 specified weights.

In the case of shelf 1, we note that geometrically the initial and final configurations are not very different. Even then, the validity-restoring path is non-trivial to find by trial and error, especially since there are different interactions involving simultaneous rotation and anisotropic scaling of multiple components. In the case of shelf 2, the top and the big side planks get adjusted over the course of the guided exploration to result in a shape that can withstand the three vertical loads. Note that the complexity of the configuration space rapidly grows with the number of planks, making it increasingly difficult to design valid shapes manually without computational support and guidance. We observe that while it is possible to restore validity by using thicker planks with more weight-bearing capacity (see Figure 6.20), this unfortunately results in higher cost, lower efficiency, and unnecessarily bulky designs. Table 6.1 presents typical continuous and discrete suggestion generation times. For generating suggestions, we can explore $\sum_{k=1,2,3} \binom{N_\gamma}{k} = O({N_\gamma}^3)$ directions in real-time even for 15-20 planks using linear approximation with the line search step taking the majority of the time. We recall that each additional plank increases N_γ by roughly 8 (see Section 6.5.1).

6.6.1 Validation of Nail Joint Physics Model

We validated the durability of our nail-joint model (see Equation 6.12) in a simple cantilever beam example with the same material assumed in our system. We observe that the maximum weight cantilever beam closely follows our model. Analogous to yielding, we observed a rapid increase in shape deformation around the maximum predicted weight.

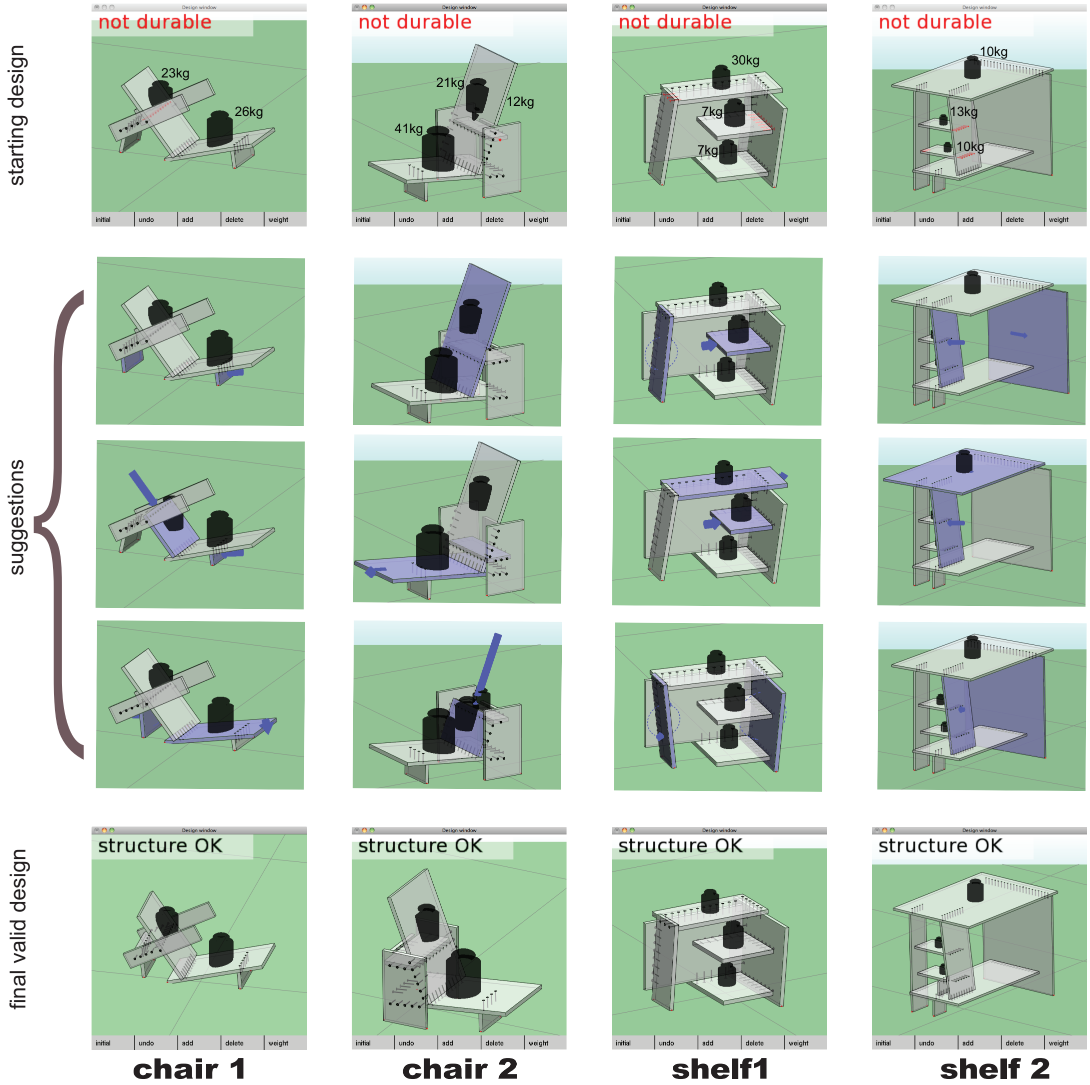


Figure 6.19: *Typical nail-jointed furniture design sessions in our guided exploration framework. Only a few suggestions are shown in each example. Note that suggestions often involve synchronous manipulation of multiple planks, which is difficult to perform without computational support.*

We performed a simple experiment to validate the nail joint model (equation 4 in our paper). We created cantilever beam (in the Figure 6.21). The cantilever beam consists of two planks and one nail joint. We hang weights on the beam and measured displacement at the tip. We increased the weight gradually and, at each weight, we also unloaded the weight and measure the displacement. This displacement in unloaded state corresponds to the amount of slip between the nails undergo during the loading process.

Three nails supported the cantilever beam. Each nails has $32mm$ length and can bear $35kN/m$ pulling force per length. The plank was $12mm$ thickness. Each of nails goes into plank $32mm - 12mm = 20mm = 0.02m$. From Equation 6.12 in our paper, maximum bending moment the joint can support is $0.02 \times 35000 \times 0.006 \times 3 =$

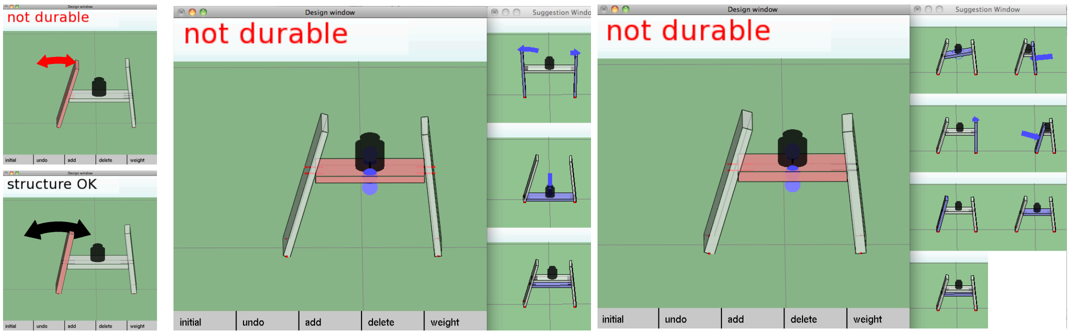


Figure 6.20: *Effect of plank thickness. Increasing the plank's thickness leads to a larger valid shape space and more suggestions.*

$1.86Nm$. The cantilever has arm length $0.19m$. Hence in the model, the cantilever can bear $1.86/0.19 = 0.97N = 1kg$.

The Figure 6.22 shows the result of actual experiment. When weights exceed the thresholds predicted by our software, the displacement under load increased rapidly out of linear curve, and also the displacement after unloading become obvious, showing the nails are slipping against plank and joint is collapsing.

6.6.2 User study

We performed a user study to obtain feedback from users. Recreating a typical design scenario, we asked the user to design a piece of furniture freely following their own concept. Nine test users (novice designers, graduate students of computer science department, one female) designed furniture with three types of systems: (i) a system without feedback from the physical simulation (i.e., the participants were not informed about which joints were undurable or whether the furniture toppled during

Table 6.1: *Performance statistics on a laptop computer with an Intel Core^{TM}i7 2.8GHz CPU with 4GB RAM.*

	Figure 6.19 right	Figure 6.18 right	Figure 6.18 middle	Figure 6.2 right
#planks	9	10	20	28
#joints	13	13	33	49
#continuous suggestion	8	8	8	6
candidate generation (ms)	13.2	22.3	160	758
line search (ms)	92.3	83.1	670	1512
#discrete suggestion	1	1	2	1
discrete suggestion (ms)	3.8	5.6	48	52
total time (ms)	110	123	880	2420

Figure 6.21: *Our experiment settings. At the tips of the nail jointed cantilever, we hang weights and measured the displacement at the tips. The Left figure shows the case of light weight (about $300g$), the cantilever could still bear the weight. The right figure shows the case of heavy weight (about $1.5kg$), the cantilever's nail yielded to the weight.*

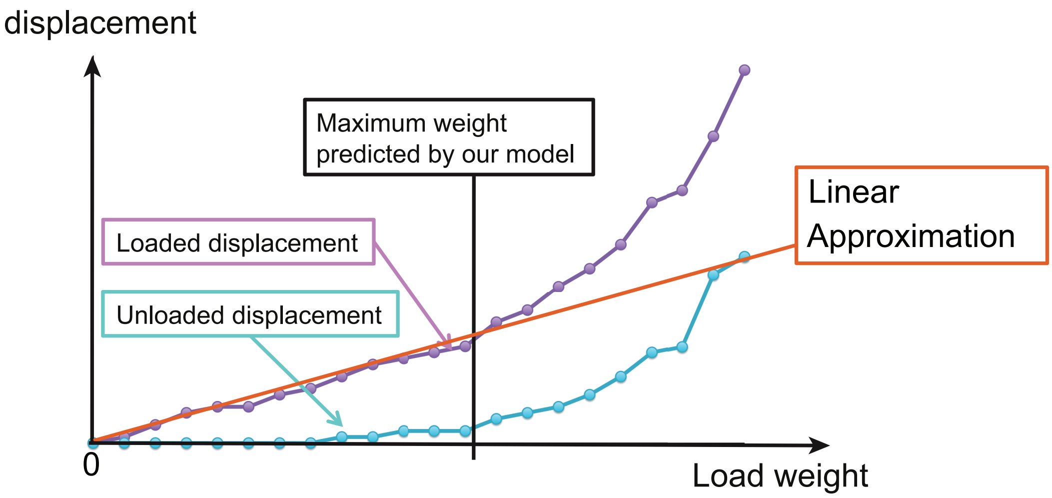


Figure 6.22: *The deformation of the cantilever beam with respect to the amount of applied weight. When the amount of weight exceeds the maximum predicted by our nail-joint physics model, the loaded deformation deviate from the linear approximation. The unloaded deformation, which is measured unloaded status in each increase of weight, suddenly takes non zero value, implicating nail joint's slip.*

design) and without suggestions, (ii) a system with feedback from the simulation (validity check and valid range visualization) but without suggestions, and (iii) a system with feedback and suggestions (our system). While using system (i), the user was allowed to see the result of simulation up to five times whenever she liked, assuming the use of a traditional shape modeling software along with a simulation software. Each participant started by creating a concept design on paper. Then she created 3D furniture models with the three systems to realize their concept design. To counter-balance learning effects, we separated the nine participants into two groups: five participants used the system in the order (i, ii, iii), while the rest used the system in the reversed order (iii, ii, i). On an average, the participants took roughly 30 minutes per successful design. We present session histories in the supplementary materials.

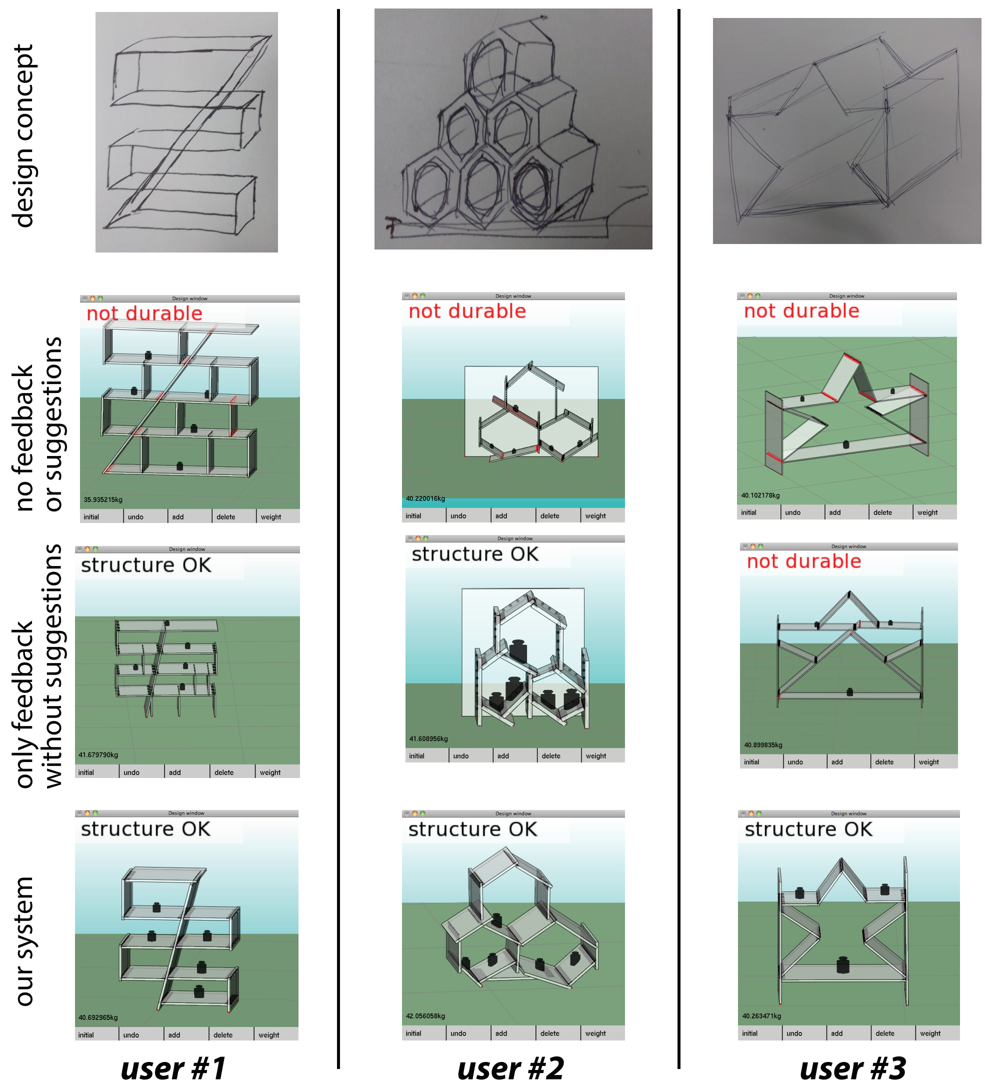


Figure 6.23: *Starting from a design concept (top row), three failed attempts with no feedback or suggestions (second row), using only feedback without suggestions (third row), and results using our system (bottom row).*

In Figure 6.23, we show a session where the participant, who was proficient at

Google SketchUp, used the systems in order (i, ii, iii). The participant simply failed to design a valid shape using system (i). With system (ii), he managed to design a valid piece of furniture, but he complained that the shape of the furniture was boring and far from his initial design concept. Using system (iii), he successfully designed a valid piece of furniture closely following his initial concept. Other participants had a similar experience (see supplementary materials for the other user sessions). All nine participants successfully created valid pieces of furniture close to their initial concepts with our system. Note that even the participants who used the system in the order (iii, ii, i) mostly failed to recreate the design with system (ii) or (i) although they had seen successful designs while using system (iii). Intuitively a validity-restoring suggestion often involves synchronous editing of multiple parts, which is challenging without suitable computational support. A participant commented that displaying the range of edits was very useful for fine-tuning a design. We note that a more rigorous quantitative comparative study of such creative design support is needed.

6.7 Limitations and Future Work

In this chapter, we presented an interactive computational design framework for guided exploration of physically valid shapes for nail-jointed furniture. Our system provides active real-time guidance to the user to help her avoid invalid designs, either due to stability violations, or due to excessive joint bending forces. We proposed a novel force-space analysis for both bending forces and frictional constraints to generate multiple suggestions, along with valid deformation ranges, involving both continuous and discrete geometric changes. We used our system to design a range of furniture and also demonstrated the utility of the system by building a physical prototype.

6.7.1 Limitations

There are still many limitations due to the various assumptions we have made. We consider planks to be perfectly rigid and unbreakable. In practice, however, planks deform under heavy loads, influencing their nail-joint behavior and ultimately they can break. This is especially true in the context of shelves or other furniture with long segments without any supporting structures. We also do not consider curved planks or shifting loads in our framework. Further, our linear approximations for computing durability and stability constraints can be violated in highly non-linear regions. Although it is possible to consider higher-order approximations, we decided against such a choice in favor of interactivity. Finally, we restricted $M = 3$, thus limiting the range of design possibilities. In certain cases, it is desirable to explore the range of meaningful suggestions especially in designs with many components, or when the initial design is far from the valid space.

Exploring valid design spaces is difficult, especially when the constraints are nonlinear. While characterizing the valid space itself is difficult, exploring high degrees of freedom design spaces is challenging as the valid regions maybe disjoint forming islands or have narrow connection pathways among valid spaces, posing further challenges. In such cases, our technique can fail to find durable configurations, even when they exist. Finally, we do not consider aesthetics in our framework. Ideally, aesthetic considerations should come from designers while our goal is simply to computationally assist the form-finding process by guiding the designer away from invalid or uninteresting parts of the shape space.

6.7.2 Future Work

A lot remains unexplored in this area. In the future, we want to ensure validity for dynamic furniture, e.g., designing a physically valid rocking chair. A possible approach is to treat the problem as a coupled exploration of multiple shapes based on the contact points to the ground and the relative (upright) orientation of the shape. Subsequently, we can simultaneously explore the multiple shapes, while adding a regularity term to favor edits that are consistent across all shapes (since correspondence is known). Finally, we plan to support exploration of shape design involving a large number of components, e.g., designing a building, etc.

Chapter 7

Conclusion

In this thesis, we have addressed the problem of designing physically valid objects without domain-specific knowledge by integrating real-time physical simulations into interactive design systems. This chapter contains a brief summary and discussion of the work, together with some possibilities for future research.

7.1 Summary

In the traditional approaches to designing physically valid objects, a simulation is performed after the design is finished, and the design is then iteratively modified until the simulation results are satisfactory. However, it is difficult for users to improve a design by investigating such offline simulation results, because the relationship between the design and the simulation results is not clear to novice users without domain-specific knowledge. To address this problem, we developed a method for integrating real-time simulation into interactive design. The simulation is performed *during* the interactive design phase, and the system provides constant feedback on the physical properties. The real-time feedback from the simulation provides guidance for better designs, and the user can explore design principles by interacting with the design. The user can also design physically valid objects more easily with a system that actively displays simulation results as suggestions and annotations. When simulation results are presented in real time during the shape-editing phase, the user can improve the shape intuitively. In this thesis, we introduced three algorithms to support such integration of the design process and real-time simulation:

- Reuse of redundant intermediate data
- First-order approximation
- Force space analysis

These algorithms were designed to quickly display the physical properties of a current design during interactive design. We demonstrated the effectiveness of this approach by implementing various design systems.

7.1.1 Reuse of Redundant Intermediate Data

We demonstrated an acceleration technique for a finite element simulation response to changes in rest shape by reusing redundant data that undergo little or no modification. We investigated the finite element procedure and identified which data can be reused. The traditional FEM framework runs a simulation from scratch whenever the rest shape changes, whereas we generate the mesh and data structures inside FEM and the linear system solvers. In our approach, the mesh is deformed to fit the altered geometry as the rest shape changes, reducing the cost of mesh generation. The internal data structure of the FEM is also reused according to the level of the mesh change. This algorithm was applied to custom metallophone design systems. Each of the metallophone pieces vibrates with specific eigenfrequencies to produce tones. It has hitherto been difficult for a user to design a metallophone, because the relationship between the shape and the eigenfrequencies is not obvious. Using our resizing algorithm, we demonstrated that the system can carry out eigenanalysis in real time, allowing users to design metallophones with arbitrary shapes.

7.1.2 First-Order Approximation

We leveraged the continuity of the simulation response to obtain a quick, scalable response in static FEM simulations. Because the response of a finite element simulation changes smoothly with respect to the rest shape, first-order approximation is reasonable. The design is changed by direct user manipulation, using a two-dimensional (2D) input device. Hence, only two degrees of freedom are necessary to parameterize a design change: the changes in the x- and y-coordinates of the pointing device during shape editing. The system pre-computes the first-order approximation with respect to the mouse movement when the user begins dragging. The first-order approximation is computed via a technique called design-sensitivity analysis, which investigates the relationship between the design and the simulation results in the case of a static solution. Because the approximation is expressed only as a linear combination of the two modes, the response is quick and scalable to the simulation size. With the warm start obtained from first-order approximation, a nonlinear simulation converges quickly. We demonstrated the effectiveness of this acceleration technique with an example involving clothing pattern design. Every article of clothing is made from 2D fabric and designed according to a pattern. The pattern specifies how to cut the fabric and stitch the pieces together to fit to a 3D human body. Because the fabric is 2D and the resulting article of clothing is 3D, the relationship between the pattern and the clothing is not clear, especially to those who do not have special knowledge of pattern design. Our system supports a user's clothing design by providing interactive feedback from a static clothing simulation during pattern editing.

7.1.3 Force Space Analysis

We explored an algorithm for quickly producing suggestions and annotations by investigating physical constraints on the force domain, rather than the geometric domain. Many physical constraints are written as relationships between forces. For example, whether or not a material yields is given by the amount of stress, and whether or not a structure collapses is determined by the sign of the contact force with the ground. We considered the force space, in which forces act as degrees of freedom. A physically valid design can be considered a region in the force space. Force space analysis is used to investigate how force configurations change with the design. We use force space analysis with a first-order approximation to display the ranges of parameters as annotations during design editing, and generate suggestions to restore structural soundness. Determining the range and generating suggestions both require quick estimation of a stable solution. In determining the range, we use estimation to decide the step size in the bisection method. Estimation is also used to generate candidates for the suggestions. Force space analysis quickly provides approximations of stable designs, thus allowing annotations and suggestions to be actively presented to users. This algorithm was demonstrated with plank-based, nail-joined furniture design. The furniture is constructed from planks (i.e., wooden boards) and nail joints. We imposed two physical constraints: stability (the furniture stands without falling over) and durability (each nail can support a certain weight without collapsing). The user designs a piece of furniture by direct manipulation. During shape editing, the system displays information on the structure ' s stability and durability in real time by running a physical simulation in the background. For each direct manipulation, the system presents a valid parameter range as an annotation. If stability or durability is violated, the system suggests several physically valid structures. These suggestions and annotations are generated quickly via the force space analysis. With these suggestions and annotations, users successfully designed stable and durable furniture.

7.2 Future Directions

There are several possible directions for further investigating the integration of interactive modeling and real-time physical simulation, on both the application side and the algorithm side. We first describe the general directions of interest, and then we discuss two problem-specific future lines of research in some detail.

7.2.1 Exploration of Dynamic Shapes

In this thesis, we described two algorithms: first-order approximation and force space analysis. Because these two algorithms can only be used if the simulation is static, we would like to develop a technique capable of handling dynamic behavior. Static equilibrium is required for predicting a linear response to design change. However, dynamic behaviors have too many degrees of freedom, and it is difficult to determine their relationship with a design. Hence, we need to find a way to characterize dynamic behaviors with a few degrees of freedom, using modes or key frames to explore their relationship with a design. Some dynamic behaviors are history-dependent, i.e., they depend on how the objects deform. Such history dependency is difficult to handle, because a small perturbation in the initial state may evolve into a large difference. Many types of deformations, such as plasticity and fractures, are also history-dependent. Thus, extending our algorithm to cover history-dependent motions is one of the most challenging directions for future work.

7.2.2 Precomputation

One of the most interesting approaches for further accelerating a computation is intensive precomputation. The precomputation approach splits the overall computations into two parts, the precomputation stage and the runtime stage, so that performance is maximized at runtime. In the design of physically valid shapes, it might be possible for the system to compute many different possible shapes beforehand, and simply present a precomputed result at runtime. The speed of the internet and the cost efficiency of data centers increase from one year to the next. Hence, we can leverage the high speed of the internet and low storage costs to store precomputed data online and download it readily when required. We are interested in (i) how to separate computations into precomputation and computation at runtime, (ii) how to represent precomputed data in a database to maximize the efficiency of storage, and (iii) how to compress the data transfer from the precomputed database so that the precomputed data are downloaded and displayed quickly.

7.2.3 Exploration of Pareto Fronts

Another future direction is designing objects that simultaneously satisfy multiple objectives. For example, in furniture design, we have to consider various objectives, such as structural soundness, material cost, and construction cost. In many cases, two different objectives have a trade-off relationship: if one property is optimized, the other property becomes sub-optimized. Multiple optimization functions produce a set of optimal solutions, called a Pareto front. Hence, developing an algorithm that allows the user to explore Pareto fronts intuitively is another possible direction for future research.

7.2.4 Accurate Large-Scale Simulations

The simulation presented in this thesis was on a relatively small scale. The accuracy and speed of a simulation have a trade-off relationship. Hence, it is necessary to explore more efficient integration schemes to obtain more accurate simulations of complex objects. One approach yet to be explored is the multi-resolutional or hierarchal approach, which considers parts of an object substructures, and solves the total system as an assembly of substructures. Another interesting approach for large-scale simulations is the dimensional reduction technique [9] for geometric models used in simulations. In this approach, structural elements such as shell elements or beam elements are leveraged to increase the efficiency of the discretization.

7.2.5 Application to Other Design Problems

On the application side, this thesis presented three specific examples: metallophone, clothing, and furniture design. However, the algorithms themselves are general, and could be used in many other design applications. There are numerous objects that people are interested in designing, including tents (in which the structure is supported by an elastic beam and a thin membrane), hairstyles (using physical hair simulations), wind instruments such as ocarinas or flutes, foldable furniture, upholstery (cushions, sofas), and paper airplanes that actually fly (using aerodynamics simulations).

7.2.6 Toward the Interactive Design of a Metallophone with Overtones

In Section 4.4, we described the design of metallophone plates with specific eigenfrequencies. This metallophone design system handles only the lowest eigenfrequency, while neglecting all overtones, and thus it cannot predict the timbre of a metallophone. Overtones were difficult to compute in real time, because the computational cost was high when using the Responsive FEM for eigenanalysis described in Appendix A. The speed of the responsive FEM is limited by the time required to compute a single iteration of the FEM eigenanalysis. The computation of multiple eigenvalues becomes slower according to the number of eigenvalues, and hence real-time simulation of overtones was infeasible. However, it may be possible to accelerate the computation of multiple eigenvalues by using the acceleration technique described in Chapter 5. The vibration of a metallophone plate is non-stationary, and thus the plate is considered a moving object. However, we can assume modal vibration, in which the vibration is decomposed into a number of constant eigenmodes. These eigenmodes are constant during vibration, and hence we can carry out the eigenanalysis as a static FEM simulation. In Appendix D, we described how to compute the sensitivity of an FEM eigenmode $\tilde{\mathbf{u}}$ and eigenvalue ω with respect to a change in a design parameter γ:

$$\frac{\partial \omega}{\partial \gamma} = \frac{\tilde{\mathbf{u}}^T \left(\mathbf{K}' - \omega^2 \mathbf{M}'\right) \tilde{\mathbf{u}}}{2\omega \tilde{\mathbf{u}}^T \mathbf{M} \tilde{\mathbf{u}}}, \tag{7.1}$$

$$\frac{\partial \tilde{\mathbf{u}}}{\partial \gamma} = \left[\mathbf{K} - \omega^2 \mathbf{M} + \epsilon \tilde{\mathbf{u}} \tilde{\mathbf{u}}^T\right]^{-1} \left\{ \tilde{\mathbf{v}} - \frac{\tilde{\mathbf{u}}^T \tilde{\mathbf{v}}}{2\omega \tilde{\mathbf{u}}^T \mathbf{M} \tilde{\mathbf{u}}} \mathbf{M} \tilde{\mathbf{u}} \right\}, \tag{7.2}$$

where $\mathbf{K}$ is a stiffness matrix, $\mathbf{M}$ is a mass matrix, $\mathbf{K}' = \partial \mathbf{K} / \partial \gamma$, $\mathbf{M}' = \partial \mathbf{M} / \partial \gamma$, and $\tilde{\mathbf{v}} = \left(\mathbf{K}' - \omega^2 \mathbf{M}'\right) \tilde{\mathbf{u}}$.

Before the user starts to design a metallophone, the system first predicts the eigenvalues and eigenmodes. Multiple eigenvalues are approximated, using standard FEM eigenanalysis techniques such as the Lanczos algorithm [140]. Then the eigenvalue and its corresponding eigenmodes are computed accurately via the inverse-power method [140], starting from the approximate eigenvalue. The inverse-power method predicts an eigenvalue and its eigenmodes sequentially, from the smallest eigenvalue to the higher ones. We can accelerate the inverse-power method by leveraging the orthogonality of the eigenmodes. When we compute an eigenvalue, we project its eigenmode so that it becomes orthogonal to the lower eigenmodes during the iteration of the procedure, as described in Appendix D. The sensitivities of the eigenvalues and eigenmodes are evaluated when the user begins to edit the shape of the metallophone. Once these sensitivities have been computed, the system can provide fast first-order predictions with a speed independent of the scalability of the problem and the number of eigenvalues. During the user ' s shape editing, the system also iterates the inverse-power method described in the previous paragraph. The progressive nonlinear augmentation techniques described in Section 5.1.5 would allow the system to predict a nonlinear frequency response The application of real-time large-scale FEM eigenanalysis during shape editing is not limited to metallophone design. In mechanical design, we usually consider buckling when a compressive force is applied to rod-shaped mechanical parts. Buckling is usually modeled via Euler buckling, in which the eigenvalue of a structure predicts the minimum amount of load required to collapse the structure. In architectural design, vibrations of a structure around specific frequencies are always considered. The vibrations may be induced by an earthquake or a Karman vortex sheet caused by strong winds. Many types of design employ eigenanalysis, and could therefore benefit from this acceleration technique.

7.2.7 Toward Guided Exploration of Physically Valid 3D Solid Objects

The most important future direction for our research for guided exploration method is to support the modeling of 3D elastic objects. More specifically, we aim to achieve a real-time validity check and the generation of annotations and suggestions for 3D objects, subject to the physical constraint that the objects are only deformable within their material limits. Here, we briefly describe a potential method that could be applied to such a problem. Application to 3D solid objects would be very important in the field of mechanical engineering, as most objects in this field (such as machinery components) are designed as 3D solid models. Thus, 3D CAD systems for engineering design and most CAE systems for engineering design run FEM solid mechanics simulations. However, as was mentioned in Chapter 2, the design of 3D objects based on physical properties is generally very difficult when using existing design systems (in which design and simulation are separated). The most significant difference between a 3D solid elastic model and the furniture model discussed in this chapter is the number of constraints. In the furniture design problem, we evaluate whether or not a piece of furniture collapses only in terms of the nail joints, assuming that all planks are rigid and undeformable. Thus, the nail joints are considered the only parts of the furniture that can be deformed. On the other hand, the deformation of a 3D solid object is continuously distributed. Any part of such an object can be deformed, and thus we must check validity everywhere inside the 3D shape. Physical constraints pertaining to object fracture are represented in the form of equivalent stress. We typically employ the von Mises yield criterion in the case of ductile materials, and the maximum principle stress in the case of brittle materials. We evaluate these fracture criteria in every FEM element to check the validity of the 3D shape. A stress in a 3D object is represented as a 3×3 symmetric tensor, and has six free parameters. The fracture criteria in a single element can be represented using these parameters. Thus, the force space has a dimensionality of six times the number of elements. The durable region in the force space of a 3D shape is a direct product of the durable regions of the individual elements, described by fracture criteria. For suggestion generation, we determine a combination of linearized trajectories, as in the case of nail-jointed furniture. The number of elements can be enormous, resulting in a force space with extraordinarily large dimensionality. We would need considerable acceleration to compute such a large-scale problem in a reasonable amount of time. One possible way to accomplish this is to reduce the dimensionality of the force space. Because the stress tensor generally varies smoothly inside an object, the stress tensors of neighboring elements do not vary independently. Thus, we can assume that the stress field would be well represented by a linear combination of smooth basis fields. Such fields can be generated using the asymptotic expansion of the singular function defined around the object ’ s surface. With these basis fields, we can significantly reduce the dimensionality of the force space, and thereby speed up the force space analysis.

The integration of interactive shape editing and real-time physical simulation is a new topic for both computer graphics and engineering design. This thesis demonstrated its potential for the creation of physically valid objects by novice users, with convincing algorithms and solid implementations. However, there are still many issues that remain to be explored in the development of tools for the creation of functional objects by novice users. We believe our thesis serves as a first step toward a significant milestone in this direction.

References

[1] Autodesk 123D. Autodesk. *(http://www.123dapp.com/)*.

[2] Abacus. Dassault systemes. *(http://www.3ds.com/products/simulia/)*.

[3] Joël Frelat Adrien Mamou-Mani and Charles Besnainou. Numerical simulation of a piano soundboard under downbearing. *Journal of the Acoustical Society of America*, 123(4):2401–2406, January 2008.

[4] Marc Alexa and Wojciech Matusik. Reliefs as images. *ACM Transaction on Graphics (SIGGRAPH 2010)*, 29:60:1–60:7, July 2010.

[5] Jérémie Allard, François Faure, Hadrien Courtecuisse, Florent Falipou, Christian Duriez, and Paul G. Kry. Volume contact constraints at arbitrary resolution. *ACM Transaction on Graphics*, 29:82:1–82:10, July 2010.

[6] Ariel Shamir Daniel Cohen-Or Andrei Sharf, Marina Blumenkrants. Snap-paste: An interactive technique for easy mesh composition. *Pacific Graphics*, 2006.

[7] Ansys. Ansys.inc. *(http://www.ansys.com/)*.

[8] ANSYS. Designxplorer. *(http://www.ansys.com/)*.

[9] Cecil G. Armstrong. Modelling requirements for finite-element analysis. *Computer-Aided Design*, 26(7):573–578, 1994.

[10] SN Atluri, JY Cho, and H.G. Kim. Analysis of thin beams, using the meshless local Petrov–Galerkin method, with generalized moving least squares interpolations. *Computational Mechanics*, 24(5):334–347, 1999.

[11] David Baraff and Andrew Witkin. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 43–54, New York, NY, USA, 1998. ACM.

[12] Jernej Barbič, Marco da Silva, and Jovan Popović. Deformable object animation using reduced optimal control. *ACM Transaction on Graphics (SIGGRAPH 2009)*, 28(3):53:1–53:9, July 2009.

[13] Jernej Barbič and Doug James. Real-time subspace integration for st. venant-kirchhoff deformable models. *ACM Transaction on Graphics (SIGGRAPH 2005)*, 24(3):982–990, July 2005.

[14] Jernej Barbič and Doug James. Time-critical distributed contact for 6-dof haptic rendering of adaptively sampled reduced deformable models. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '07, pages 171–180, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[15] Jernej Barbič and Jovan Popović. Real-time control of physically based simulations using gentle forces. *ACM Transaction on Graphics (SIGGRAPH Asia 2008)*, 27(5):163:1–163:10, December 2008.

[16] Jernej Barbič and Yili Zhao. Real-time large-deformation substructuring. *ACM Transaction on Graphics (SIGGRAPH 2011)*, 30(4):91:1–91:8, July 2011.

[17] MR Barnes. Form-finding and analysis of prestressed nets and membranes. *Computers & structures*, 30(3):685–695, 1988.

[18] Klaus-Jurgen Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice Hall, February 1982.

[19] Jean-Louis Batoz, Klaus-JÜRgen Bathe, and Lee-Wing Ho. A study of three-node triangular plate bending elements. *International Journal for Numerical Methods in Engineering*, 15(12):1771–1812, 1980.

[20] Christopher Batty, Florence Bertails, and Robert Bridson. A fast variational framework for accurate solid-fluid coupling. *ACM Transaction on Graphics (SIGGRAPH 2007)*, 26(3), July 2007.

[21] Eliane Bécache, Antoine Chaigne, Gregoire Derveaux, and Patrick Joly. Numerical simulation of a guitar. *Comput. Struct.*, 83(2-3):107–126, January 2005.

[22] Richard Bergman. *Wood Handbook – Wood as an Engineering Material*. Forest Products Laboratory, 2010.

[23] Miklos Bergou, Max Wardetzky, David Harmon, Denis Zorin, and Eitan Grinspun. A quadratic bending model for inextensible surfaces. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 227–230, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[24] 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 Transaction on Graphics (SIGGRAPH 2010)*, 29(4):63:1–63:10, July 2010.

[25] Zdenek Bittnar and Jiri Sejnoha. *Numerical Methods in Structural Mechanics*. American Society of Civil Engineers, June 1996.

[26] blender.org. Blender. *(http://www.blender.org/)*.

[27] Jozsef Bodig and Benjamin A. Jayne. *Mechanics of Wood and Wood Composites*. Krieger Pub Co, 5 1993.

[28] Martin Bokeloh, Michael Wand, and Hans-Peter Seidel. A connection between partial symmetry and inverse procedural modeling. *ACM Transaction on Graphics (SIGGRAPH 2010)*, 29(4):104:1–104:10, July 2010.

[29] Javier Bonet and Richard D. Wood. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, 1997.

[30] Javier Bonet and Richard D. Wood. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, 2 edition, March 2008.

[31] Folkmar A. Bornemann and Peter Deuflhard. The cascadic multigrid method for elliptic problems. *Numerische Mathematik*, 75:135–152, 1996. 10.1007/s002110050234.

[32] P. Breitkopf, H. Naceur, A. Rassineux, and P. Villon. Moving least squares response surface approximation: formulation and metal forming applications. *Computers & Structures*, 83(17-18):1411–1428, 2005.

[33] J. Bretos and C. Santamara. Finite element analysis and experimental measurements of natural eigenmodes and random responses of wooden bars used in musical instruments. *Applied Acoustics*, 56(3):141–156, 1999.

[34] Robert Bridson, Sebastian Marino, and Ronald Fedkiw. Simulation of clothing with folds and wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '03, pages 28–36, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[35] Felippa CA. Dynamic relaxation in quasi-newton methods. *Numerical methods for nonlinear problems,*, pages 27–38, 1986.

[36] Mantilla Carrasco Carla Paoliello, Edgar Vladimiro. Chair load analysis during daily sitting activities. *Forest Products Journal*, 58(9), 2008.

[37] CATIA. Dassault systemes. *(http://www.3ds.com/products/catia)*.

[38] Siddhartha Chaudhuri, Evangelos Kalogerakis, Leonidas Guibas, and Vladlen Koltun. Probabilistic reasoning for assembly-based 3d modeling. *ACM Transaction on Graphics (SIGGRAPH 2011)*, 30(4):35:1–35:10, July 2011.

[39] Siddhartha Chaudhuri and Vladlen Koltun. Data-driven suggestions for creativity support in 3D modeling. *ACM Transaction on Graphics (SIGGRAPH Asia 2010)*, 29(6):183:1–183:10, december 2010.

[40] Stephen Chenney and D. A. Forsyth. Sampling plausible solutions to multi-body constraint problems. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 219–228, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[41] Nuttapong Chentanez, Ron Alterovitz, Daniel Ritchie, Lita Cho, Kris K. Hauser, Ken Goldberg, Jonathan R. Shewchuk, and James F. O'Brien. Interactive simulation of surgical needle insertion and steering. *ACM Transaction on Graphics (SIGGRAPH 2009)*, 28(3):88:1–88:10, August 2009.

[42] Lam Yee Cheong, Deng Yimin, Tor Shu Beng, S. B. Tor, and Graeme A. Britton. Feature-based cad-cae integration model for injection-moulded product design. *International Journal of Production Research*, 40(15):3737–3750, 2002.

[43] Kwang-Jin Choi and Hyeong-Seok Ko. Stable but responsive cloth. In *SIGGRAPH*, pages 604–611, 2002.

[44] Kwang-Jin Choi and Hyeong-Seok Ko. Research problems in clothing simulation. *Computer-Aided Design*, 37(June):585 – 592, 2005.

[45] Lucas F.M. da Silva, Paulo J.C. das Neves, R.D. Adams, and J.K. Spelt. Analytical models of adhesively bonded joints — part i: Literature survey. *International Journal of Adhesion and Adhesives*, 29(3):319 – 330, 2009.

[46] L. Daudeville, L. Davenne, and M. Yasumura. Prediction of the load carrying capacity of bolted timber joints. *Wood Science and Technology*, 33:15–29, 1999. 10.1007/s002260050095.

[47] A S Day. An introduction to dynamic relaxation. *The Engineer*, 219:218–221, January 1965.

[48] Edilson de Aguiar, Leonid Sigal, Adrien Treuille, and Jessica K. Hodgins. Stable spaces for real-time clothing. *ACM Transaction on Graphics (SIGGRAPH 2010)*, 29(4):106:1–106:9, July 2010.

[49] 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*, 25(3):625–634, 2006.

[50] Herve Delingette. Triangular springs for modeling nonlinear membranes. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):329–341, 2008.

[51] Alexandre Derouet-Jourdan, Florence Bertails-Descoubes, and Joëlle Thollot. Stable inverse dynamic curves. *ACM Transaction on Graphics (SIGGRAPH 2010)*, 29(6):137:1–137:10, December 2010.

[52] Reader's Digest. *New Complete Guide to Sewing (Readers Digest)*. Readers Digest, October 2002.

[53] Michael Eigensatz, Martin Kilian, Alexander Schiftner, Niloy J. Mitra, Helmut Pottmann, and Mark Pauly. Paneling architectural freeform surfaces. *ACM Transaction on Graphics (SIGGRAPH 2010)*, 29(4):45:1–45:10, July 2010.

[54] Kenny Erleben, Jon Sporring, Knud Henriksen, and Henrik Dohlmann. *Physics Based Animation (Graphics Series)*. Charles River Media, August 2005.

[55] Fab@Home. A fab@home fabber model i. *(http://fabathome.org/)*.

[56] Raanan Fattal and Dani Lischinski. Target-driven smoke animation. *ACM Transaction on Graphics (SIGGRAPH 2004)*, 23(3):441–448, August 2004.

[57] Wei-Wen Feng, Yizhou Yu, and Byung-Uck Kim. A deformation transformer for real-time cloth animation. *ACM Transaction on Graphics (SIGGRAPH 2010)*, 29(4):108:1–108:9, July 2010.

[58] Neville H. Fletcher and Thomas D. Rossing. *The Physics of Musical Instruments*. Springer, 2nd ed. 1998. corr. 5th printing edition, 6 1998.

[59] M. Michael Floater. Mean value coordinates. *Computer Aided Geometric Design*, 20(1):19–27, 2003.

[60] Marzia Fontana, Caterina Rizzi, and Umberto Cugini. A cad-oriented cloth simulation system with stable and efficient ode solution. *Computers & Graphics*, 30(3):391 – 406, 2006.

[61] Center for Bits and Atoms at MIT. Fablab. *(http://fab.cba.mit.edu)*.

[62] T.P. Fries and H.G. Matthies. Classification and overview of meshfree methods. *Department of Mathematics and Computer Science, Technical University of Braunschweig*, 2003.

[63] Sarah F. Frisken, Ronald N. Perry, Alyn P. Rockwood, and Thouis R. Jones. Adaptively sampled distance fields: a general representation of shape for computer graphics. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 249–254, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[64] Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin. Modeling by example. *ACM Transaction on Graphics (SIGGRAPH 2004)*, 23(3):652–663, August 2004.

[65] Ran Gal, Olga Sorkine, Niloy J. Mitra, and Daniel Cohen-Or. iWIRES: an analyze-and-edit approach to shape manipulation. *ACM Transaction on Graphics (SIGGRAPH 2009)*, 28(3):33:1–33:10, July 2009.

[66] Michel Geradin and Alberto Cardona. *Flexible Multibody Dynamics: A Finite Element Approach*. Wiley, 1 edition, February 2001.

[67] Neil Gershenfeld. *FAB: The Coming Revolution on Your Desktop–From Personal Computers to Personal Fabrication*. Basic Books, April 2005.

[68] Yotam Gingold, Takeo Igarashi, and Denis Zorin. Structured annotations for 2d-to-3d modeling. *ACM Transaction on Graphics (SIGGRAPH Asia 2009)*, 28(5):148:1–148:9, December 2009.

[69] Rony Goldenthal, David Harmon, Raanan Fattal, Michel Bercovier, and Eitan Grinspun. Efficient simulation of inextensible cloth. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, 26(3), July 2007.

[70] Google. Sketchup. *http://sketchup.google.com/*.

[71] Eitan Grinspun, Anil Hirani, Mathieu Desbrun, and Peter Schröder. Discrete Shells. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pages 62–67, August 2003.

[72] Havok.Inc. Havok. *(http://www.havok.com/)*.

[73] Miloš Hašan, Martin Fuchs, Wojciech Matusik, Hanspeter Pfister, and Szymon Rusinkiewicz. Physical reproduction of materials with specified subsurface scattering. *ACM Transaction on Graphics (SIGGRAPH 2010)*, 29:61:1–61:10, July 2010.

[74] Michiel Hazewinkel, editor. *Encyclopaedia of Mathematics (Set*. Kluwer Academic Pub, reprint edition, February 2000.

[75] Kristian Hildebrand, Bernd Bickel, and Marc Alexa. crdbrd: Shape fabrication by sliding planar slices. In *to appear: Computer Graphics Forum (Eurographics 2012)*, volume 31, 2012.

[76] Stefan Holmberg, Kent Persson, and Hans Petersson. Nonlinear mechanical behaviour and analysis of wood and fibre materials. *Computers & Structures*, 72(4-5):459 – 480, 1999.

[77] Hugues Hoppe. Progressive meshes. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 99–108, New York, NY, USA, 1996. ACM.

[78] Takeo Igarashi and John F. Hughes. A suggestive interface for 3D drawing. In *UIST*, pages 173–181, 2001.

[79] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: a sketching interface for 3d freeform design. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 409–416, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[80] Takeo Igarashi, Tomer Moscovich, and John F. Hughes. As-rigid-as-possible shape manipulation. *ACM Transaction on Graphics*, 24(3):1134–1141, July 2005.

[81] IKEA. Ikea planner tools. *(http://www.ikea.com/)*.

[82] Xavier Provot Institut and Xavier Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *In Graphics Interface*, pages 147–154, 1996.

[83] Hiroshi Ishii and Brygg Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '97, pages 234–241, New York, NY, USA, 1997. ACM.

[84] ISO/DIS. Furniture-chairs-determination of stability. *Draft International Standard*, 7174(1), 1979.

[85] Doug L. James and Dinesh K. Pai. Artdefo: accurate real time deformable objects. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 65–72, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[86] Doug L. James and Dinesh K. Pai. Dyrt: dynamic response textures for real time deformation simulation with graphics hardware. *ACM Transaction on Graphics*, 21(3):582–585, July 2002.

[87] Olga A. Karpenko and John F. Hughes. Smoothsketch: 3d free-form shapes from complex sketches. *ACM Transaction on Graphics (SIGGRAPH 2006)*, 25(3):589–598, July 2006.

[88] Xavier Boutillon Kerem Ege. Vibration and acoustical characteristics of the piano soudboard. In *Proceedings of 20th International Congress on Acoustics(ICA 2010)*, 2010.

[89] William B. Kerr and Fabio Pellacini. Toward evaluating material design interface paradigms for novice users. *ACM Transaction on Graphics (SIGGRAPH 2010)*, 29(4):35:1–35:10, July 2010.

[90] Peyman Khosravi, Rajamohan Ganesan, and Ramin Sedaghati. Corotational non-linear analysis of thin plates and shells using a new shell elemen. *IJNME*, 69:859–885, 2007.

[91] J. Kindel and I.C. Wang. Vibrations of a piano soundboard: Modal analysis and finite element analysis. *The Journal of the Acoustical Society of America*, 81:S61, 1987.

[92] A. Klarbring. Examples of non-uniqueness and non-existence of solutions to quasistatic contact problems with friction. *Archive of Applied Mechanics*, 60:529–541, 1990.

[93] Vladislav Kraevoy, Alla Sheffer, Ariel Shamir, and Daniel Cohen-Or. Non-homogeneous resizing of complex models. *ACM Transaction on Graphics*, 27(5):111:1–111:9, December 2008.

[94] François Labelle and Jonathan Richard Shewchuk. Isosurface stuffing: fast tetrahedral meshes with good dihedral angles. *ACM Transaction on Graphics*, 26(3):57, July 2007.

[95] Vanchai Laemlaksakul. Innovative design of laminated bamboo furniture using finite element method. *INTERNATIONAL JOURNAL OF MATHEMATICS AND COMPUTERS IN SIMULATION*, 3(2), 2008.

[96] Manfred Lau, Akira Ohgawara, Jun Mitani, and Takeo Igarashi. Converting 3d furniture models to fabricatable parts and connectors. *ACM Transaction on Graphics*, 30(4):85:1–85:6, July 2011.

[97] Sang Hun Lee. A cad-cae integration approach using feature-based multi-resolution and multi-abstraction modelling techniques. *Comput. Aided Des.*, 37(9):941–955, August 2005.

[98] Antoine Chaigne Leïla Rhaouti and Patrick Joly. Time-domain modeling and numerical simulation of a kettledrum. *Journal of the Acoustical Society of America*, 105(6):3545–3562, February 1999.

[99] Xian-Ying Li, Tao Ju, Yan Gu, and Shi-Min Hu. A geometric study of v-style pop-ups: theories and algorithms. *ACM Transaction on Graphics*, 30(4):98:1–98:10, July 2011.

[100] Xian-Ying Li, Chao-Hui Shen, Shi-Sheng Huang, Tao Ju, and Shi-Min Hu. Popup: automatic paper architectures from 3d models. *ACM Transaction on Graphics*, 29(4):111:1–111:9, July 2010.

[101] Bullet Physics Library. Bullet. *(http://bulletphysics.org/)*.

[102] Yaron Lipman, Johannes Kopf, Daniel Cohen-Or, and David Levin. Gpu-assisted positive mean value coordinates for mesh deformations. In *Proceedings of the fifth Eurographics symposium on Geometry processing*, SGP '07, pages 117–123, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[103] Yang Liu, Helmut Pottmann, Johannes Wallner, Yong-Liang Yang, and Wenping Wang. Geometric modeling with conical meshes and developable surfaces. *ACM Transaction on Graphics*, 25(3):681–689, July 2006.

[104] Kui-Yip Lo, Chi-Wing Fu, and Hongwei Li. 3d polyomino puzzle. *ACM Transactino on Graphics*, 28(5):157:1–157:8, December 2009.

[105] R. E. Rowlands M. U. Rahman, Y. J. Chiang. Stress and failure analysis of double-bolted joints in douglas-fir and sitka spruce. *Wood and Fiber Science*, 23(4):567:589, October 1991.

[106] Jaroslav Mackerle. Finite element analyses in wood research: a bibliography. *Wood Science and Technology*, 39:579–600, 2005.

[107] MakerBot. Makerbot replicator$^{\mathrm{TM}}$. *(http://www.makerbot.com/)*.

[108] Lawrence E. Malvern. *Introduction to the mechanics of a continuous medium*. Prentice-Hall series in engineering of the physical sciences. Prentice-Hall, Englewood Cliffs, N.J., 1969.

[109] Sebastian Martin, Peter Kaufmann, Mario Botsch, Eitan Grinspun, and Markus Gross. Unified simulation of elastic rods, shells, and solids. *ACM Transaction on Graphics (ACM SIGGRAPH 2010)*, 29(4):39:1–39:10, July 2010.

[110] Teresa De Martino, Bianca Falcidieno, and Stefan HaÃŸinger. Design and engineering process integration through a multiple view intermediate modeller in a distributed object-oriented system environment. *Computer-Aided Design*, 30(6):437 – 452, 1998.

[111] Masry Masry and Hod Lipson. A sketch-based interface for iterative design and analysis of 3d objects. *ACM SIGGRAPH 2007 courses*, 2007.

[112] James McCrae, Karan Singh, and Niloy J. Mitra. Slices: a shape-proxy based on planar sections. *ACM Transaction on Graphics*, 30(6):168:1–168:12, December 2011.

[113] Antoine McNamara, Adrien Treuille, Zoran Popović, and Jos Stam. Fluid control using the adjoint method. *ACM Transactions on Graphics (SIGGRAPH 2004)*, 23(3):449–456, August 2004.

[114] Ullrich Meier, Oscar López, Carlos Monserrat, Mari C. Juan, and Mariano Alcañiz. Real-time deformable models for surgery simulation: a survey. *Computer Methods and Programs in Biomedicine*, 77(3):183 – 197, 2005.

[115] Paul Merrell, Eric Schkufza, and Vladlen Koltun. Computer-generated residential building layouts. *ACM Transaction on Graphics (SIGGRAPH Asia 2010)*, 29(6):181:1–181:12, December 2010.

[116] Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, and Vladlen Koltun. Interactive furniture layout using interior design guidelines. *ACM Transaction on Graphics (SIGGRAPH 2011)*, 30(4):87:1–87:10, July 2011.

[117] Mark Meyer, Gilles Debunne, Mathieu Desbrun, and Alan H. Barr. Interactive animation of cloth-like objects in virtual reality. *J. Vis. Comput. Animat.*, 12(1):1–12, February 2001.

[118] Johannes Mezger, Bernhard Thomaszewski, Simon Pabst, and Wolfgang Straßer. Interactive physically-based shape editing. In *Proceedings of the 2008 ACM symposium on Solid and physical modeling*, pages 79–89, 2008.

[119] Jun Mitani and Hiromasa Suzuki. Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Transaction on Graphics (ACM SIGGRAPH 2004)*, 23(3):259–263, August 2004.

[120] Niloy J. Mitra and Mark Pauly. Shadow art. *ACM Transaction on Graphics (SIGGRAPH Asia 2009)*, 28(5):156:1–156:7, December 2009.

[121] Yuki Mori and Takeo Igarashi. Plushie: an interactive design system for plush toys. *ACM Transaction on Graphics (SIGGRAPH 2007)*, 26(3), July 2007.

[122] Matthias Müller, Julie Dorsey, Leonard McMillan, Robert Jagnow, and Barbara Cutler. Stable real-time deformations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '02, pages 49–54, New York, NY, USA, 2002. ACM.

[123] Ahmet Celal APAY Mustafa Hilmi ÇOLAKOĞLU. Finite element analysis of wooden chair strength in free drop. *International Journal of the Physical Sciences*, 7(7):1105–1114, February 2012.

[124] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Fibermesh: designing freeform surfaces with 3d curves. *ACM Transaction on Graphics (SIGGRAPH 2007)*, 26(3), July 2007.

[125] Andrew Nealen, Matthias Müller, Richard Keiser, Eddy Boxerman, and Mark Carlson. Physically based deformable models in computer graphics. *Computer Graphics Forum*, 25(4):809–836, 2006.

[126] Trevor Nicholls and Romulus Crisan. Study of the stress–strain state in corner joints and box-type furniture using finite element analysis (fea). *European Journal of Wood and Wood Products*, 60:66–71, 2002. 10.1007/s00107-001-0262-0.

[127] H. Nishigaki, S. Nishiwaki, T. Amago, and N. Kikuchi. First order analysis for automotive body structure design. In *ASME DETC*, 2000.

[128] NVIDIA. Physx. *(http://www.geforce.com/hardware/technology/physx)*.

[129] NX. Simens. *(http://www.plm.automation.siemens.com/products/nx/)*.

[130] OptiTex. Optitex. *(http://www.optitex.com/)*.

[131] Maks Ovsjanikov, Wilmot Li, Leonidas Guibas, and Niloy J. Mitra. Exploration of continuous variability in collections of 3D shapes. *ACM Transaction on Graphics (SIGGRAPH 2011)*, 30(4):33:1–33:10, July 2011.

[132] Patrick Paczkowski, Min H. Kim, Yann Morvan, Julie Dorsey, Holly Rushmeier, and Carol O'Sullivan. Insitu: sketching architectural designs in context. *ACM Transaction on Graphics (SIGGRAPH Asia 2011)*, 30(6):182:1–182:10, December 2011.

[133] Eric G. Parker and James F. O'Brien. Real-time deformation and fracture in a game environment. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '09, pages 165–175, New York, NY, USA, 2009. ACM.

[134] Harry Parker and James Ambrose. *Simplified Design of Wood Structures*. Wiley, 1997.

[135] Jovan Popović, Steven M. Seitz, Michael Erdmann, Zoran Popović, and Andrew Witkin. Interactive manipulation of rigid body simulations. pages 209–217, 2000.

[136] Helmut Pottmann, Yang Liu, Johannes Wallner, Alexander Bobenko, and Wenping Wang. Geometry of multi-layer freeform structures for architecture. *ACM Transaction on Graphics*, 26(3), July 2007.

[137] Helmut Pottmann, Alexander Schiftner, Pengbo Bo, Heinz Schmiedhofer, Wenping Wang, Niccolo Baldassini, and Johannes Wallner. Freeform surfaces from single curved panels. *ACM Transaction on Graphics*, 27(3):76:1–76:10, August 2008.

[138] Alec Rivers, Frédo Durand, and Takeo Igarashi. 3d modeling with silhouettes. *ACM Transaction on Graphics*, 29(4):109:1–109:8, July 2010.

[139] Damien Rohmer, Tiberiu Popa, Marie-Paule Cani, Stefanie Hahmann, and Alla Sheffer. Animation wrinkling: augmenting coarse cloth simulations with realistic-looking wrinkles. *ACM Transaction on Graphics*, 29(6):157:1–157:8, December 2010.

[140] Youcef Saad. *Numerical Methods for Large Eigenvalue Problems (Algorithms and Architectures for Advanced Scientific Computing)*. Manchester University Press, April 1995.

[141] Youcef Saad. *Iterative methods for sparse linear systems*. SIAM, Philadelphia, 2nd ed edition, 2003.

[142] A. Schoofs, F. Van Asperen, P. Maas, and A. Lehr. A Carillon of Major-Third Bells. I. Computation of Bell Profiles using Structural Optimization. *Music Perception*, 4(3):255–266, 1987.

[143] Shapeways. Shapeways. *(http://www.shapeways.com/)*.

[144] Lior Shapira, Ariel Shamir, and Daniel Cohen-Or. Image appearance exploration by model-based navigation. pages 629–638, 2009.

[145] Mayank Singh and Scott Schaefer. Triangle surfaces with discrete equivalence classes. *ACM Transaction on Graphics (SIGGRAPH 2010)*, 29(4):46:1–46:7, July 2010.

[146] Mélina Skouras, Bernhard Thomaszewski, Bernd Bickel, and Markus Gross. Computational design of rubber balloons. *Comput. Graphics Forum (Proc. Eurographics)*, 2012.

[147] J. Smardzewski. Strength of profile-adhesive joints. *Wood Science and Technology*, 36:173–183, 2002. 10.1007/s00226-001-0131-3.

[148] Jeffrey Smith, Jessica K Hodgins, Irving Oppenheim, and Andrew Witkin. Creating models of truss structures with optimization. 21(1):295 – 301, 2002.

[149] MSC Software. Nastran. *(http://www.mscsoftware.com/)*.

[150] Optimal Solutions Software. Sculptor. *(http://gosculptor.com/)*.

[151] Jos Stam. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 121–128, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[152] Ivan E. Sutherland. Sketchpad: a man-machine graphical communication system. In *Proceedings of the May 21-23, 1963, spring joint computer conference*, AFIPS '63 (Spring), pages 329–346, New York, NY, USA, 1963. ACM.

[153] Jerry O. Talton, Daniel Gibson, Lingfeng Yang, Pat Hanrahan, and Vladlen Koltun. Exploratory modeling with collaborative design spaces. *ACM Transaction on Graphics (SIGGRAPH Asia 2009)*, 28:167:1–167:10, December 2009.

[154] Joseph Teran, Eftychios Sifakis, Geoffrey Irving, and Ronald Fedkiw. Robust quasistatic finite elements and flesh simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '05, pages 181–190, New York, NY, USA, 2005. ACM.

[155] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. *SIGGRAPH Comput. Graph.*, 21(4):205–214, August 1987.

[156] T. E. Tezduyar, S. Mittal, S. E. Ray, and R. Shih. Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. *Comput. Methods Appl. Mech. Eng.*, 95(2):221–242, 1992.

[157] Stephen P. Timoshenko, James M. Gere, and Engineering. *Theory of Elastic Stability (Dover Civil and Mechanical Engineering)*. Dover Publications, 2 edition, June 2009.

[158] Adrien Treuille, Andrew Lewis, and Zoran Popović. Model reduction for real-time fluids. *ACM Transaction on Graphics (SIGGRAPH 2006)*, 25(3):826–834, July 2006.

[159] Adrien Treuille, Antoine McNamara, Zoran Popović, and Jos Stam. Keyframe control of smoke simulations. *ACM Transaction on Graphics*, 22(3):716–723, July 2003.

[160] Christopher D. Twigg and Doug L. James. Many-worlds browsing for control of multibody dynamics. *ACM Transaction on Graphics (SIGGRAPH 2007)*, 26(3), July 2007.

[161] Christopher D. Twigg and Doug L. James. Backwards steps in rigid body simulation. *ACM Transaction on Graphics (SIGGRAPH 2008)*, 27(3):25:1–25:10, August 2008.

[162] Christopher D. Twigg and Zoran Kačić-Alesić. Optimization for sag-free simulations. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '11, pages 225–236, New York, NY, USA, 2011. ACM.

[163] Nobuyuki Umetani, Danny M. Kaufman, Takeo Igarashi, and Eitan Grinspun. Sensitive couture for interactive garment modeling and editing. *ACM Transaction on Graphics (SIGGRAPH 2011)*, 30(4):90:1–90:12, July 2011.

[164] P Underwood. Dynamicrelaxation: a review. *Computational methods for transient dynamic analysis*, 1983.

[165] F. Van Keulen, RT Haftka, and NH Kim. Review of options for structural design sensitivity analysis. Part 1: Linear systems. *Computer Methods in Applied Mechanics and Engineering*, 194(30-33):3213–3243, 2005.

[166] Alla Sheffer Vladislav Kraevoy, Dan Julius. Shuffler: Modeling with interchangeable parts. *Pacific Graphics*, 2007.

[167] Pascal Volino, Frederic Cordier, and Nadia Magnenat-Thalmann. From early virtual garment simulation to interactive fashion design. *Comput. Aided Des.*, 37(6):593–608, May 2005.

[168] Pascal Volino and Nadia Magnenat-Thalmann. Stop-and-go cloth draping. *The Visual Computer*, 23(9-11):669–677, 2007.

[169] Pascal Volino, Nadia Magnenat-Thalmann, and Francois Faure. A simple approach to nonlinear tensile stiffness for accurate cloth simulation. *ACM Transaction on Graphics*, 28(4):105:1–105:16, September 2009.

[170] Huamin Wang, Florian Hecht, Ravi Ramamoorthi, and James O'Brien. Example-based wrinkle synthesis for clothing animation. *ACM Transaction on Graphics (SIGGRAPH 2010)*, 29(4):107:1–107:8, July 2010.

[171] Tim Weyrich, Pieter Peers, Wojciech Matusik, and Szymon Rusinkiewicz. Fabricating microgeometry for custom surface reflectance. *ACM Transaction on Graphics (SIGGRAPH 2009)*, 28(3):32:1–32:6, July 2009.

[172] Emily Whiting, John Ochsendorf, and Frédo Durand. Procedural modeling of structurally-sound masonry buildings. *ACM Transaction on Graphics (SIGGRAPH Asia 2009)*, 28(5):112, December 2009.

[173] Martin Wicke, Matt Stanton, and Adrien Treuille. Modular bases for fluid dynamics. *ACM Transaction on Graphics (ACM SIGGRAPH 2009)*, 28(3):39:1–39:8, July 2009.

[174] Shiqing Xin, Chi-Fu Lai, Chi-Wing Fu, Tien-Tsin Wong, Ying He, and Daniel Cohen-Or. Making burr puzzles from 3d models. *ACM Transaction on Graphics (SIGGRAPH 2011)*, 30(4):97:1–97:8, July 2011.

[175] Yong-Liang Yang, Yi-Jun Yang, Helmut Pottmann, and Niloy J. Mitra. Shape space exploration of constrained meshes. *ACM Transaction on Graphics (SIGGRAPH Asia 2011)*, 30(6), December 2011.

[176] Lap-Fai Yu, Sai-Kit Yeung, Chi-Keung Tang, Demetri Terzopoulos, Tony F. Chan, and Stanley J. Osher. Make it home: automatic optimization of furniture arrangement. *ACM Transaction on Graphics (SIGGRAPH 2011)*, 30(4):86:1–86:12, July 2011.

[177] Jun Mitani Yuki Igarashi, Takeo Igarashi. Interactive beadwork design and construction. *ACM Transaction on Graphics (SIGGRAPH 2012)*, 31(4), Augest 2012.

[178] Robert C. Zeleznik, Kenneth P. Herndon, and John F. Hughes. Sketch: an interface for sketching 3d scenes. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, New York, NY, USA, 2006. ACM.

Appendix A

Detail of Eigen Analysis of Metallophone

Here is a detailed derivation of the algorithm we use. The eigenvalue problem is formulated using the FEM discretization as

$$\bar{\mathbf{M}}\ddot{\boldsymbol{u}} + \mathbf{K}\boldsymbol{u} \quad = \quad \mathbf{0}, \tag{A.1}$$

where $\boldsymbol{u}$ is the nodal displacement vector, $\bar{\mathbf{M}}$ is the lumped mass matrix, and $\mathbf{K}$ is the positive semi-definite stiffness matrix. By splitting the displacement $\boldsymbol{u}$ into the product of a spatially varying amplitude $\boldsymbol{\phi}$ and a harmonic oscillation with angular velocity ω, $\boldsymbol{u}(\boldsymbol{x}, t) = \boldsymbol{\phi}(\boldsymbol{x})e^{i\omega t}$, and substituting this into Equation A.1, we obtain

$$\mathbf{K}\boldsymbol{\phi} = \lambda\bar{\mathbf{M}}\boldsymbol{\phi}, \tag{A.2}$$

where the eigenfrequency $f = \omega/(2\pi) = \sqrt{\lambda}/(2\pi)$. Our goal is to calculate the smallest nonzero eigenvalue λ and its corresponding eigenvector $\boldsymbol{\phi}$.

Cholesky factorization is employed to represent the lumped mass matrix as $\bar{\mathbf{M}} = \mathbf{L}\mathbf{L}^{\mathbf{T}}$, and both sides of Equation A.2 are multiplied on the left by $\mathbf{L}^{-1}$ to obtain the standard eigenvalue problem

$$\mathbf{A}\boldsymbol{\psi} = \lambda\boldsymbol{\psi}, \tag{A.3}$$

where $\mathbf{A} = \mathbf{L}^{-1}\mathbf{K}\mathbf{L}^{-\mathbf{T}}$ and $\boldsymbol{\psi} = \mathbf{L}^{\mathbf{T}}\boldsymbol{\phi}$. We solve this by inverse iteration. The standard iteration procedure is modified by adding a step that removes all zero eigenvectors of $\mathbf{A}$ from the current solution. Because of our problem setting, we already know that the zero eigenvectors of $\mathbf{K}$ are $\boldsymbol{\phi}_0^i$ $(i = 1, \ldots, 6)$, the translations along the three coordinate axes and the rotations around them. We then apply modified Gram-Schmidt orthogonalization to $\mathbf{L}^{\mathbf{T}}\boldsymbol{\phi}_0^i$ $(i = 1, \ldots, 6)$ to obtain the orthonormal basis vectors $\boldsymbol{\psi}_0^i$ $(i = 1, \ldots, 6)$ that span the kernel of $\mathbf{A}$, and define a projection $\mathcal{P}$ that maps a vector $\boldsymbol{v}$ to the complement space of the kernel of $\mathbf{A}$ by $\mathcal{P}(\boldsymbol{v}) = \boldsymbol{v} - \sum \boldsymbol{\psi}_0^i \left(\boldsymbol{\psi}_0^i \cdot \boldsymbol{v}\right)$. In each step of the iteration, we apply this projection to the solution vector and normalize it. We add a small positive number ε to the diagonals of $\mathbf{A}$ to improve the numerical conditions. Once the shifted eigenvalue λ_1' and its corresponding eigenvector $\boldsymbol{\psi}_1$ of $\mathbf{A}$ are computed, we finally obtain the smallest nonzero eigenvalue $\lambda_1 = \lambda_1' - \varepsilon$ and the eigenvector $\boldsymbol{\phi}_1 = \mathbf{L}^{-\mathbf{T}}\boldsymbol{\psi}_1$. Reusing of the solution from the previous configuration significantly improves the convergence of the inverse iterations. The processing returns to the main thread in each iteration step to avoid freezing to user input.

Appendix B

Detail of Membrane Physics Modeling

For the fast simulation, we modify existing formulation of three nodes constant strain triangle (CST) membrane element developed by P.Volino [169]. Volino took warp and woof as an orthogonal basis vector to represent in plainer stress. Instead, we use two triangle edges as basis vectors. This basis vectors are not orthonormal; the two edges are not in right angle and are not in unit length. However it leads simpler formulation and resulting simulation become faster. Figure B.1 shows the configuration of basis vectors. The positions of undeformed and deformed triangle's corners are written as $\mathbf{P_1 P_2 P_3}$ and $\mathbf{p_1 p_2 p_3}$. We assume $\mathbf{P_1}$ is the origin of a generalized coordinate and let the vector from origin to $\mathbf{P_1}$ and $\mathbf{P_2}$ as basis vector $\mathbf{G_1}$ and $\mathbf{G_2}$. $\mathbf{G_3}$ is defined as an unit vector orthogonal to both $\mathbf{G_1}$ and $\mathbf{G_2}$. We define $\mathbf{g_1}, \mathbf{g_2}, \mathbf{g_3}$ similarly in the deformed configuration.

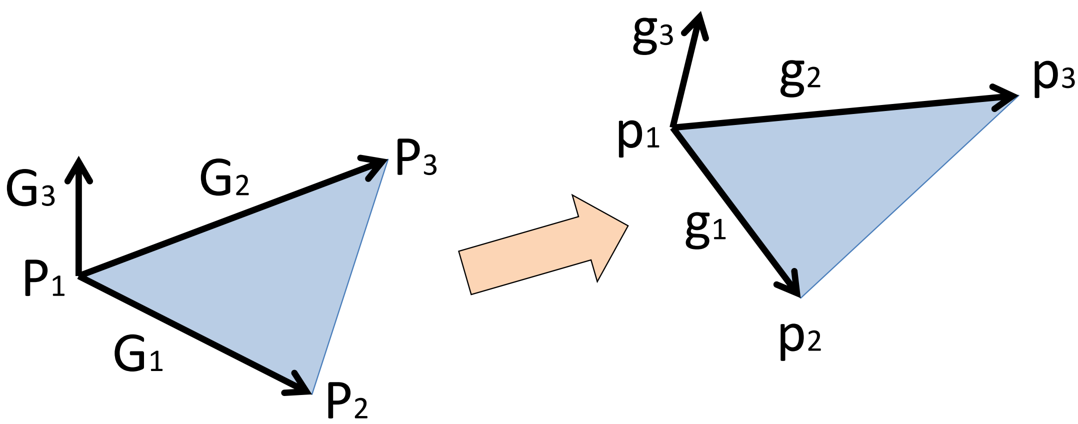


Figure B.1: *A triangle in (left) undeformed and (right) deformed configurations.*

We assume this thin shell is made out of St.Venant Kirchhoff (StVK) material, where the 2nd Piola-Kirchhoff (PK2) stress tensor $\mathbf{S}$ and the Green-Lagrange strain tensor $\mathbf{E}$ is related linearly. In current implementation, we assume the thin shell is isotropic. In such a case, the PK2 stress can be written as $\mathbf{S} = \lambda\,(\mathrm{tr}\mathbf{E})\,\mathbf{I} + \mathbf{2}\mu\mathbf{E}$ using

Lamé's constant λ and μ. The Green-Lagrange strain can be written [30] as:

$$\mathbf{E} = \frac{\mathbf{1}}{\mathbf{2}} \left(\mathbf{g_i} \cdot \mathbf{g_j} - \mathbf{G_i} \cdot \mathbf{G_j} \right) \mathbf{G^i G^{j T}}, \tag{B.1}$$

where the vectors with superscript $\mathbf{G^k}$ denote the vectors is *contra-variant*; that is the following relationship $\mathbf{G_i} \cdot \mathbf{G^j} = \delta_\mathbf{i}^\mathbf{j}$ holds where the δ_i^j is a Kronecker's delta.

The strain energy densitiy of the StVK material is an inner product of PK2 stress and Green-Lagrange strain $\mathbf{S} : \mathbf{E}$. We calculate the internal force vector and stiffness matrix by taking first and second derivative of strain energy with respect to deformed node location.

Here we define matrix $[B]$ which relates strain to nodal displacement

$$[B_{gh}]_i^p = \mathbf{g}_g^T \frac{\partial \mathbf{g}_h}{\partial \mathbf{u}_i^p} = \mathbf{g}_{gi} \frac{\partial \mathbf{N}^p}{\partial \mathbf{r}_h}, \tag{B.2}$$

where N and r is natural coordinate and its variable. The internal force of element $\mathbf{Q}_e$, which is derivative of element strain energy $We = 1/2 \int_{V_e} \mathbf{S} : \mathbf{E} dV$ w.r.t. nodal displacement $\mathbf{u}$, becomes

$$Q_{e}{}_i^p = \int_{v_e} S^{gh} \frac{\partial E_{gh}}{\partial u_i^p} dV = S^{gh} \left[B_{gh} \right]_i^p A, \tag{B.3}$$

where A is a area of this triangle element and $\mathbf{S}$ is a symmetry 2x2 PK2 stress tensor. The element stiffness matrix is derivative of element internal force $\mathbf{Q}$ w.r.t. nodal displacement $\mathbf{u}$

$$\{K_e\}_{ij}^{pq} = \left[\frac{\partial S^{gh}}{\partial u_j^q} [B_{gh}]_i^p] + S^{gh} \frac{\partial [B_{gh}]_i^p}{\partial u_j^q} \right] A \tag{B.4}$$

$$= \left[\bar{C}^{efgh} [B_{ef}]_j^q [B_{gh}]_i^p + S^{gh} \frac{\partial N^q}{\partial r_g} \frac{\partial N^p}{\partial r_h} \delta_{ij} \right] A, \tag{B.5}$$

where $\bar{C}^{efgh}$ is a symmetric part of constitutive matrix, which relate S^{gh} to E_{ef}. The first term of Equation B.4 is constant with fixed reference configuration and always positive definite. Meanwhile the property of second term of Equation B.4 is totally dependent on the stress tensor $\mathbf{S}$. We perform eigen- decomposition of stress $\mathbf{S} = \lambda_1 \mathbf{e}_1 \mathbf{e}_1^T + \lambda_2 \mathbf{e}_2 \mathbf{e}_2^T$ and calculate $\mathbf{S}^+ = \lambda_1^+ \mathbf{e}_1 \mathbf{e}_1^T + \lambda_2^+ \mathbf{e}_2 \mathbf{e}_2^T$, here λ_1^+ and λ_2^+ are λ_1 and λ_2 if they are positive and 0 if they are negative. Substituting $\mathbf{S}^+$ instead of $\mathbf{S}$ into Equation B.4 guarantees the positive definiteness of the Jacobian matrix.

Appendix C

Sensitivity Analysis Under the Constraint

Here we explain how the design sensitivity analysis is computed in a simulation of static deformation including constraints. Here we consider an initial configuration $\mathbf{\Gamma}$ undergoes deformation $\mathbf{\Lambda}$. The design sensitivity analysis calculates how the deformation $\mathbf{\Lambda}$ changes with respect to initial configuration $\mathbf{\Gamma}$. The constraint Φ was imposed on the deformation and the constraint itself changes according to the change of initial configuration. The static simulation solves a minimization problem with constraints

$$\begin{cases} minimize\ K(\mathbf{\Gamma},\mathbf{\Lambda})\ for\ all\ \mathbf{\Lambda} \\ while\ \Phi(\mathbf{\Gamma},\mathbf{\Lambda}) = 0 \end{cases}, \tag{C.1}$$

where K denotes potential energies of the system such as strain energy. Using the Lagrange multiplier method, the solution makes the function $K + \lambda\Phi$ take extremal value, where where λ is a Lagrange multiplier. Hence the following relationship holds

$$\delta(K + \lambda\Phi) = \frac{\partial K}{\partial \mathbf{\Lambda}}\delta\mathbf{\Lambda} + \lambda\frac{\partial \Phi}{\partial \mathbf{\Lambda}}\delta\mathbf{\Lambda} + \delta\lambda\Phi = 0 \ \ \forall\delta\mathbf{\Lambda}\ and\ \forall\delta\lambda. \tag{C.2}$$

This formulation can be solved with following linear system

$$\begin{bmatrix} \frac{\partial K}{\partial\mathbf{\Lambda}\partial\mathbf{\Lambda}} + \frac{\partial\Phi}{\partial\mathbf{\Lambda}\partial\mathbf{\Lambda}} & \left(\frac{\partial\Phi}{\partial\mathbf{\Lambda}}\right)^T \\ \frac{\partial\Phi}{\partial\mathbf{\Lambda}} & 0 \end{bmatrix} \begin{pmatrix} \mathbf{\Lambda} \\ \lambda \end{pmatrix} = \begin{pmatrix} \frac{\partial K}{\partial\mathbf{\Lambda}} + \frac{\partial\Phi}{\partial\mathbf{\Lambda}} \\ -\Phi \end{pmatrix}. \tag{C.3}$$

Here we consider sensitivity by introducing infinitesimal perturbation into $\mathbf{\Gamma}$ and $\mathbf{\Lambda}$ in the relationship given by Equation C.2 as:

$$\begin{aligned} &\frac{\partial}{\partial\mathbf{\Lambda}}K\left(\mathbf{\Gamma}+\Delta\mathbf{\Gamma},\mathbf{\Lambda}+\Delta\mathbf{\Lambda}\right) \\ &\qquad + (\lambda+\Delta\lambda)\frac{\partial}{\partial\mathbf{\Lambda}}\Phi\left(\mathbf{\Gamma}+\Delta\mathbf{\Gamma},\mathbf{\Lambda}+\Delta\mathbf{\Lambda}\right) \simeq 0 \end{aligned} \tag{C.4}$$

$$\begin{aligned} \Leftrightarrow\ \ &\frac{\partial}{\partial\mathbf{\Lambda}}\left[K + \frac{\partial K}{\partial\mathbf{\Gamma}}\Delta\mathbf{\Gamma} + \frac{\partial K}{\partial\mathbf{\Lambda}}\Delta\mathbf{\Lambda}\right] \\ &\qquad + (\lambda+\Delta\lambda)\frac{\partial}{\partial\mathbf{\Lambda}}\left[\Phi + \frac{\partial\Phi}{\partial\mathbf{\Gamma}}\Delta\mathbf{\Gamma} + \frac{\partial\Phi}{\partial\mathbf{\Lambda}}\Delta\mathbf{\Lambda}\right] \simeq 0 \end{aligned} \tag{C.5}$$

$$\begin{aligned} \Leftrightarrow\ \ &\left(\frac{\partial K}{\partial\mathbf{\Lambda}} + \lambda\frac{\partial\Phi}{\partial\mathbf{\Lambda}}\right) + \left(\frac{\partial K}{\partial\mathbf{\Lambda}\partial\mathbf{\Gamma}} + \frac{\partial\Phi}{\partial\mathbf{\Lambda}\partial\mathbf{\Gamma}}\right)\Delta\mathbf{\Gamma} \\ &\qquad + \left(\frac{\partial K}{\partial\mathbf{\Lambda}\partial\mathbf{\Lambda}} + \frac{\partial\Phi}{\partial\mathbf{\Lambda}\partial\mathbf{\Lambda}}\right)\frac{\partial\mathbf{\Lambda}}{\partial\mathbf{\Gamma}}\Delta\mathbf{\Gamma} + \frac{\partial\Phi}{\partial\mathbf{\Lambda}}\frac{\partial\lambda}{\partial\mathbf{\Gamma}}\Delta\mathbf{\Gamma} \simeq 0. \end{aligned} \tag{C.6}$$

It leads to a linear system

$$\left[\begin{array}{cc} \frac{\partial K}{\partial \boldsymbol{\Lambda} \partial \boldsymbol{\Lambda}} + \frac{\partial \Phi}{\partial \boldsymbol{\Lambda} \partial \boldsymbol{\Lambda}} & \left(\frac{\partial \Phi}{\partial \boldsymbol{\Lambda}}\right)^T \\ \frac{\partial \Phi}{\partial \boldsymbol{\Lambda}} & 0 \end{array}\right] \left(\begin{array}{c} \frac{\partial \boldsymbol{\Lambda}}{\partial \boldsymbol{\Gamma}} \\ \frac{\partial \lambda}{\partial \boldsymbol{\Gamma}} \end{array}\right) = \left(\begin{array}{c} \frac{\partial K}{\partial \boldsymbol{\Lambda} \partial \boldsymbol{\Gamma}} + \frac{\partial \Phi}{\partial \boldsymbol{\Lambda} \partial \boldsymbol{\Gamma}} \\ -\frac{\partial \Phi}{\partial \boldsymbol{\Gamma}} \end{array}\right), \tag{C.7}$$

for solving the sensitivity. By comparing Equation C.3 and Equation C.7, we can conclude that the coefficient matrices of both linear systems is identical and the left-hand-side and right-hand-side vectors are different only in the point that they are differentiated with respect to the initial shape $\boldsymbol{\Gamma}$ in the design sensitivity formulation.

Appendix D

Sensitivity Analysis of Eigen Value and Mode

In this appendix, we describe computation of eigen-value and eigen-mode's design sensitivity. More specifically, we explain first order approximation of how eigen-value and eigen-mode of a FEM eigen-analysis varie with respect to a design change. We consider an generalized eigenvalue problem

$$\left(\mathbf{M} - \omega^2 \mathbf{K}\right) \tilde{\mathbf{u}} = 0, \tag{D.1}$$

where $\mathbf{M}$ is a symmetric positive definite mass matrix, $\mathbf{K}$ is a positive symmetric semi-definite stiffness matrix, ω is a square of an eigen value and $\tilde{\mathbf{u}}$ is a eigen-mode. Because the eigen-mode has scale redundancy, we put an unit-length constraint to the eigen-mode

$$\tilde{\mathbf{u}}^T \tilde{\mathbf{u}} = 1. \tag{D.2}$$

First, we apply perturbation to the mass matrix $\Delta\mathbf{M}$, the stiffness matrix $\Delta\mathbf{K}$, the square root of eigen value $\Delta\omega$, and the eigen-mode $\Delta\mathbf{u}$ in the Equation D.1

$$
\begin{aligned}
& \left\{(\mathbf{M} + \Delta\mathbf{M}) - (\omega + \Delta\omega)^2 (\mathbf{K} + \Delta\mathbf{K})\right\} (\tilde{\mathbf{u}} + \Delta\tilde{\mathbf{u}}) = 0, && \text{(D.3)} \\
\Leftrightarrow\ & \left\{(\mathbf{M} + \Delta\mathbf{M}) - (\omega^2 + 2\omega\Delta\omega)(\mathbf{K} + \Delta\mathbf{K})\right\} (\tilde{\mathbf{u}} + \Delta\tilde{\mathbf{u}}) = 0, && \text{(D.4)} \\
\Leftrightarrow\ & \left(\Delta\mathbf{K} - \omega^2 \Delta\mathbf{M} - 2\omega\Delta\omega\mathbf{M}\right) \tilde{\mathbf{u}} + \left(\mathbf{K} - \omega^2\mathbf{M}\right) \Delta\tilde{\mathbf{u}} = 0. && \text{(D.5)}
\end{aligned}
$$

We first solve for $\Delta\omega$ by multiplying $\tilde{\mathbf{u}}$ from left of the Equation D.5. Note that the second term of the left hand side becomes zero using Equation D.1 and symmetry of matrix $\mathbf{K}$ and $\mathbf{M}$.

$$
\begin{aligned}
& \tilde{\mathbf{u}}^T \left(\Delta\mathbf{K} - \omega^2 \Delta\mathbf{M} - 2\omega\Delta\omega\mathbf{M}\right) \tilde{\mathbf{u}} = 0, && \text{(D.6)} \\
\Leftrightarrow\ & \Delta\omega = \frac{\tilde{\mathbf{u}}^T \left(\Delta\mathbf{K} - \omega^2 \Delta\mathbf{M}\right) \tilde{\mathbf{u}}}{2\omega \tilde{\mathbf{u}}^T \mathbf{M} \tilde{\mathbf{u}}}. && \text{(D.7)}
\end{aligned}
$$

Considering infinitesimal perturbation, Equation D.6 leads to the sensitivity of the eigen-value with respect to a design parameter γ

$$\frac{\partial\omega}{\partial\gamma} = \frac{\tilde{\mathbf{u}}^T \left(\mathbf{K}' - \omega^2 \mathbf{M}'\right) \tilde{\mathbf{u}}}{2\omega \tilde{\mathbf{u}}^T \mathbf{M} \tilde{\mathbf{u}}}, \tag{D.8}$$

where $\mathbf{M}' = \partial M/\partial\gamma$ and $\mathbf{K}' = \partial K/\partial\gamma$. Note that the sensitivity of the eigen value can be compute without solving any linear systems.

Secondary, we compute sensitivity of eigen-mode. We substitute $\Delta\omega$ in Equation D.5 using Equation D.7

$$\begin{cases} \left(\mathbf{K} - \omega^2\mathbf{M}\right)\Delta\tilde{\mathbf{u}} = \mathbf{f} \\ \mathbf{f} = \left(\Delta\mathbf{K} - \omega^2\Delta\mathbf{M}\right)\tilde{\mathbf{u}} - \frac{\tilde{\mathbf{u}}^T\left(\Delta\mathbf{K}-\omega^2\Delta\mathbf{M}\right)\tilde{\mathbf{u}}}{2\omega\tilde{\mathbf{u}}^T\mathbf{M}\tilde{\mathbf{u}}}\mathbf{M}\tilde{\mathbf{u}}. \end{cases} \tag{D.9}$$

We also introduce perturbation of the unit-length constraint of the eigen-mode into the Equation D.2

$$(\tilde{\mathbf{u}} + \Delta\tilde{\mathbf{u}})^T(\tilde{\mathbf{u}} + \Delta\tilde{\mathbf{u}}) = 1, \tag{D.10}$$

$$\Leftrightarrow \quad \tilde{\mathbf{u}}^T\tilde{\mathbf{u}} + 2\tilde{\mathbf{u}}^T\Delta\tilde{\mathbf{u}} + \Delta\tilde{\mathbf{u}}^T\Delta\tilde{\mathbf{u}} = 1, \tag{D.11}$$

$$\Leftrightarrow \quad \tilde{\mathbf{u}}^T\Delta\tilde{\mathbf{u}} = 0. \tag{D.12}$$

The perturbation of the eigen-mode $\Delta\tilde{\mathbf{u}}$ have to satisfy both Eqation D.12 and Equation D.9. Note that the coefficient matrix of Equation D.9, $(\mathbf{K} - \omega^2\mathbf{M})$, is a singular matrix and has a kernel $\tilde{\mathbf{u}}$. Hence, Equation D.9 cannot be solved directly. To overcome this problem, we add $(\epsilon\tilde{\mathbf{u}}\tilde{\mathbf{u}}^T)$, where ϵ is a arbitrary positive value, inorder to make the coefficient matrix positive definite and thus invertible.

$$\left(\mathbf{K} - \omega^2\mathbf{M} + \epsilon\mathbf{u}\mathbf{u}^T\right)\Delta\tilde{\mathbf{u}} = \mathbf{f}. \tag{D.13}$$

Note that this operation doesn't change the solution $\Delta\tilde{\mathbf{u}}$ since $\mathbf{f}^T\tilde{\mathbf{u}} = 0$ holds from Equation D.9. Equation D.14 satisfies Equation D.12 automatically. This leads to the sensitivity of the eigen mode

$$\frac{\partial\tilde{\mathbf{u}}}{\partial\gamma} = \left[\mathbf{K} - \omega^2\mathbf{M} + \epsilon\mathbf{u}\mathbf{u}^T\right]^{-1}\left\{\tilde{\mathbf{v}} - \frac{\tilde{\mathbf{u}}^T\tilde{\mathbf{v}}}{2\omega\tilde{\mathbf{u}}^T\mathbf{M}\tilde{\mathbf{u}}}\mathbf{M}\tilde{\mathbf{u}}\right\}, \tag{D.14}$$

where $\tilde{\mathbf{v}} = \left(\mathbf{K}' - \omega^2\mathbf{M}'\right)\tilde{\mathbf{u}}$.