

Optimization of Natural Frequencies for Fabrication-Aware Shape Modeling

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Visual Computing

eingereicht von

Christian Hafner

Matrikelnummer 0925172

an der Fakultät für Informatik
der Technischen Universität Wien

Betreuung: Associate Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer

Mitwirkung: Projektass.(FWF) Dr.techn. Dipl.-Mediensys.wiss. Przemyslaw Musialski

Wien, 29. Juni 2015

Christian Hafner

Michael Wimmer

Optimization of Natural Frequencies for Fabrication-Aware Shape Modeling

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Visual Computing

by

Christian Hafner

Registration Number 0925172

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Associate Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer

Assistance: Projektass.(FWF) Dr.techn. Dipl.-Mediensys.wiss. Przemyslaw Musialski

Vienna, 29th June, 2015

Christian Hafner

Michael Wimmer

Erklärung zur Verfassung der Arbeit

Christian Hafner
Radetzkystraße 14/11, 1030 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 29. Juni 2015

Christian Hafner

Danksagung

Zu Anfang möchte ich herzlichst meinem Betreuer Przemyslaw Musialski dafür danken, mich an seiner Forschungsarbeit zu beteiligen, sich immer Zeit für meine Fragen zu nehmen, mir mit Rat und Tat zur Seite zu stehen, und meine Motivation hochzuhalten. Außerdem möchte ich mich bei meinem Betreuer Michael Wimmer für seine sorgfältige Kritik und zahlreichen Verbesserungsvorschläge bedanken.

Meine tiefste Dankbarkeit gebührt meinen Eltern Renate und Norbert, die mich seit jeher in allen Bestrebungen unterstützt und mir alles gegeben haben.

Zu guter Letzt danke ich meinen lieben Mitbewohnern Marta und Jeff, die unsere Wohnung zu einem Zuhause machen.

Acknowledgements

First of all, I would like to thank my supervisor Przemyslaw Musialski for affording me the opportunity to take part in his research, always making time for my inquiries, being a constant source of integral ideas, and motivating me to work hard. I would also like to thank my supervisor Michael Wimmer for his diligent feedback and for setting high standards to help make this work the best version of itself.

My deepest gratitude goes to my parents Renate and Norbert, who have supported my every ambition and who have given me everything from day one.

Last but not least, I thank my dear roommates Marta and Jeff for making our apartment feel like home.

Kurzfassung

Diese Arbeit befasst sich mit der Synthese von drei-dimensionalen Geometrien, deren natürliches Frequenzspektrum die gewünschte Zielfrequenz enthält und deren Erscheinungsbild dem beabsichtigten Entwurf ähnelt. Wir schlagen drei Methoden zur Parametrisierung von Geometrien vor, welche sinnvolle Freiheitsgrade zur Konstruktion von Musikinstrumenten wie Marimba und Glocken gewähren. Die Formfamilie, auf welcher der Optimierungsalgorithmus arbeitet, basiert auf der Repräsentation eines Körpers als das Volumen zwischen einer Außenfläche und einer inneren Versatzfläche. Die Evaluierung des Frequenzspektrums eines gegebenen Körpers wird mithilfe der Finite-Elemente-Methode vorgenommen. Zu diesem Zweck werden verschiedene Elementtypen erklärt und deren Eignung für die vorliegende Aufgabe analysiert. Zur Herstellung eines optimierten Körpers stellen wir eine Fabrikationsmethode vor, welche es Amateurhandwerkern ermöglicht, mithilfe von Sand- oder Kautschukformen Zinninstrumente herzustellen. Die Effizienz dieser Methode wird an den Beispielen einer simplen Zinnglocke und einer komplexeren Glocke in Form eines Hasen demonstriert. Die Tonhöhe der Glocken stimmt mit der von dem Finite-Elemente-Modell prognostizierten Frequenz mit einer Abweichung von 2,8% bzw. 6% überein. Wir ergänzen diese Resultate mit einer Reihe von rechentechnischen Ergebnissen, welche die Optimierung von Obertonfrequenzverhältnissen sowie den Einflussgrad von Netzauflösung und Geometrieglätte auf die Genauigkeit des Berechnungsmodells erforschen.

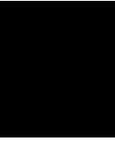
Abstract

Given a target shape and a target frequency, we automatically synthesize shapes that exhibit this frequency as part of their natural spectrum while resembling the target shape as closely as possible. We propose three shape parametrization methods that afford meaningful degrees of freedom in the design of instruments such as marimbas and bells. The design space is based on the representation of a solid as the volume enclosed by an outer surface and an inner offset surface. In order to evaluate the natural frequency spectrum of a solid, we employ finite element modal analysis and evaluate the suitability of different element types. We propose a fabrication method for the production of optimized instruments by an amateur craftsperson using sand or rubber molds. The efficiency of our method is demonstrated by the production of a simple tin bell and a more complicated bell in the shape of a rabbit. We achieve agreement with the predicted pitch frequencies of 2.8% and 6% respectively. These physical results are supplemented by a number of computational results that explore the optimization of harmonic ratios and the influence of mesh resolution and mesh smoothness on the accuracy of the finite element model.

Contents

Kurzfassung	xi
Abstract	xiii
Contents	xv
1 Introduction	1
1.1 Motivation	1
1.2 Problem Statement	2
1.3 Contributions	3
1.4 Overview	4
2 Theoretical Foundations	7
2.1 Linear Elasticity Theory	7
2.2 Virtual Work and Energy Principles	16
2.3 Finite Element Discretization	18
2.4 Solving the Equations of Motion	24
2.5 Numerical Integration	25
3 Related Works	33
3.1 Shape Optimization	33
3.2 Sound Synthesis	36
3.3 Instrument Analysis	37
4 Optimization of Natural Frequencies	39
4.1 Overview	39
4.2 Design Space	39
4.3 Finite Element Modeling	48
4.4 Non-Linear Optimization	60
5 Validation and Results	63
5.1 Validation Method	63
5.2 Results	81
	xv

6	Possible Optimizations	97
6.1	Overview	97
6.2	Mesh-Resolution Benchmarks	98
6.3	Mesh-Smoothness Benchmark	99
7	Conclusion	105
7.1	Synopsis	105
7.2	Future Work	107
	Bibliography	109



Introduction

1.1 Motivation

The computer graphics community has dedicated many of its efforts to the synthesis of images that reflect parts of the real world, be it to achieve photorealistic special effects or to visualize data in a comprehensible manner. Computers and displays are ubiquitous, which is why digital content, such as videos, games, and e-books, enjoys enormous popularity. This trend has even affected the production of tangible objects: E-books make up 30% of all book sales in the U.S. now, and digital video game sales surpassed the sale of physical copies in the U.K. in 2013.

While we can conclude that consumers see digital content as a viable alternative to mass-produced items, limited and unique products are always in great demand. In the past, creating a unique artifact has always been costly because it requires an expert designer and expensive equipment to manufacture. Techniques for quickly producing individual products have been used in industrial manufacturing processes under the umbrella term “rapid prototyping” for more than thirty years. Especially additive manufacturing techniques such as Fused Deposition Modeling (FDM) and Selective Laser Sintering (SLS) have become a constant in CAD-CAM workflows. Other industries that benefit from such techniques are the fashion, jewelry, and entertainment industries.

The latest development in additive manufacturing has largely mitigated the problem of prohibitively high costs: Cheap FDM printers have been available to a broad audience for about five years and enable even hobbyists and enthusiasts to use this technology. Emergent online portals such as Thingiverse bear testimony to the widespread popularity that desktop 3d printers have come to enjoy.

The advent of consumer-grade 3d printers has recently sparked interest in a novel research direction: fabrication-aware shape design and optimization. This exciting field leaves the realm of digital shape representations and focuses on the creation of physical artifacts. Recent publications demonstrate a multitude of applications, the interdisciplinary nature of the field, and diverse goals: reduction of printing costs,

automatic stability analysis and improvements for printed models, synthesis of models with particular mass properties, optimization of deformation properties, and many more.

In this master thesis we want to investigate a property that has not been discussed in a shape optimization setting before: the natural frequency spectrum of a three-dimensional solid. The natural frequency spectrum encodes the vibrations that an object can undergo and therefore the pitch and timbre of the sounds that it can exhibit. Since the frequency spectrum is an intrinsic property that depends on the shape and the material of an object, it can be influenced by varying the geometry. This makes the natural frequency spectrum a likely candidate for shape optimization techniques.

The most straightforward application of natural frequency optimization is the design of struck idiophones—instruments that vibrate as a whole after being struck with a mallet, like xylophones, triangles, and cymbals—with little need for user intervention. Instrument shapes do not usually deviate from well-established designs like the rectangular bars of a xylophone. The reason is that creating a novel instrument is a pain-staking trial-and-error process that necessitates fabricating prototypes in order to evaluate pitch and sound quality. Section 3.3 reviews works in acoustics that attempt to quantify relationships between instrument shape and sound in a computational manner. Most of the works in this area act in an analytical capacity and do not propose automated design solutions. The goal of this thesis is to explore computational solutions that facilitate the design of instruments by synthesizing 3d shapes to match a desired pitch.

1.2 Problem Statement

We aspire to create a workflow that helps a layperson design and fabricate an idiophone. The user should be able to control the pitch and the targeted material and have considerable influence over the shape of the finished result. Additionally, the user can set a secondary goal in the form of harmonic ratios between the first overtones.

We can identify a number of subproblems that arise on the road to achieve these goals. A fundamental question is which computational model to choose in order to establish the connection between a given 3d object and its natural frequency spectrum. This is a well-studied area in engineering because natural frequencies play an important role in civil engineering for the prevention of resonance disasters (e.g., the collapse of the Tacoma Narrows Bridge in 1940) and earthquake damages. The challenge lies in choosing a model that is predictive enough to achieve a small error compared to the real natural frequency spectrum. At the same time the evaluation of the model should take as little time as possible in order to be a suitable candidate for a shape optimization algorithm.

A physical counterpart for the computational model is needed to validate the results. Measuring the natural frequencies of a physical object is also a well-researched subject in acoustics and in civil engineering, but the most common approach, which is briefly discussed in Section 3.3, requires expensive equipment that is unavailable to us. We need to find an alternative that allows us to measure natural frequencies by means of simple tools.

The other challenge is to choose an appropriate design space for the creation of idiophones. The design space is a parametrized set of shapes that an optimization algorithm can use to search for an optimal solution. One requirement is that it be large enough to pinpoint frequencies in an adequate range, e.g., at least one octave for a xylophone-like instrument. The other requirement is that the user be able to control the shape of the instrument to a large degree. Thus, the visible parts of the instrument should deviate as little as possible from the design targeted by the user. A third requirement is that the design space only contain physically plausible shapes, i.e., the shapes must not contain any self-intersections, and it should be possible to fabricate them with the means at hand.

With a parametrized design space and a method to compute natural frequencies in place, it is necessary to formulate an optimization problem. The constraints that the formulation puts on the optimization parameters must ensure that physically implausible shapes are excluded from the search. Ideally, the optimization problem should be amenable to an off-the-shelf numerical solver.

The final problem is to choose a material and find a suitable fabrication method for an optimized idiophone. The material has to be easy to acquire, cheap, and malleable by means accessible to an amateur craftsperson.

1.3 Contributions

We provide a theoretical introduction to natural frequency analysis and explore a spectrum of solutions to the problems listed above. In particular, we use finite element modal analysis to determine the natural frequency spectrum of a three-dimensional solid computationally. To this end, we evaluate the suitability of different types of elements and compare the results for a variety of shapes. We also provide benchmarks for geometric primitives and compare our computational solutions to frequency measurements.

To validate the results found by our finite element models, we develop a poor man's version of the experimental modal analysis methods used in an engineering context. Instead of using accelerometers, we take audio samples with a commercially available microphone and extract frequency peaks in a semi-automatic fashion. This method is used to assess the predictive power of our finite element model on the examples of store-bought aluminium plates and self-produced idiophones.

In order to establish a suitable design space, we survey typical shapes and tuning methods for a variety of idiophones to identify shape parameters that influence the natural frequency spectrum. We then propose three methods of shape parametrization with varying degrees of flexibility. All three methods use the idea of defining a 3d object as the volume enclosed by an outer surface with boundary and an inner surface that has been offset from the outer surface. The first design space uses the offset distance as a parameter in order to regulate the pitch. This method is only applicable for very smooth shapes because it can easily produce self-intersecting surfaces in high-curvature regions. The second parametrization method uses the same parameter but calculates the offset inner surface in a more sophisticated way. It introduces a meso-skeleton to

limit the per-vertex offset distances in order to prevent self-intersections and intersections between the two surfaces. These two parametrizations work with a single optimization parameter and are therefore efficient in finding solutions that pinpoint a desired target pitch. However, they cannot optimize overtone frequencies for lack of degrees of freedom. This issue is addressed by the third parametrization method, which uses an adjustable number of parameters. Depending on the complexity of the shape and of secondary objectives, the number of parameters can be chosen to yield a larger design space. This method is based on an order-reduction scheme using manifold harmonics.

We propose a three-stage fabrication pipeline that allows the inexpensive production of an optimized idiophone by a layperson. Metals are suitable materials because they are isotropic, thus affording precise finite element analysis, hard, thus exhibiting good-quality sounds, and they can be made to assume almost arbitrary shapes using the casting process. This manufacturing method requires the use of a heat-resistant mold into which the molten metal is poured to acquire the desired shape. In our pipeline, we first produce a positive of our 3d object out of plastic using an FDM printer. From the positive, we create a mold using either oiled molding sand for simple shapes or silicone caoutchouc for more complicated shapes. The mold is equipped with an opening into which molten tin is poured to form the idiophone.

We demonstrate the optimization pipeline and the fabrication method by manufacturing differently shaped tin bells that exhibit a desired pitch. To assess the quality of the results, we measure the natural frequencies of the fabricated solids and compare them to the frequencies predicted by the finite element model. In order to further improve the accuracy of the prediction, we estimate the material properties of the tin we use from a set of sound samples with an optimization algorithm. The manifold harmonics parametrization is demonstrated on a computational result in which the harmonic ratio between the pitch and the first overtone of a marimba bar is optimized using a Monte Carlo sampling algorithm combined with local optimization.

The behavior of our finite element model is explored further on a series of computational benchmarks. We analyze the effects of mesh resolution on the accuracy of natural frequency calculations using the finite element method. Additionally, we analyze the influence of small surface details and mesh smoothness on the natural frequencies of an object. These experiments are carried out with the goal of speeding up future implementations of a natural frequency optimization framework.

1.4 Overview

Chapter 2 serves as an introduction to linear elasticity theory as a mathematical model to describe the motions and deformations of solid objects. It presents the concept of virtual work and the principles of potential and kinetic energy in the context of deforming bodies. These theories are the basis of the finite element approximation, which can be used to simulate the behavior of an object computationally. The result of the finite element model is a system of ordinary differential equations, whose solution is also discussed. The finite element method makes heavy use of volume integrals that have to be integrated

numerically. In the last section of this chapter, we derive the Newton–Cotes formulae and several Gauss quadrature rules that are needed to evaluate the integrals. We also discuss the problem of spurious modes, which occurs if the integrals are solved with insufficient accuracy.

Chapter 3 reviews seminal works in the disciplines that are relevant to this thesis. Shape optimization techniques have gained traction in the computer graphics community in the last five years. The most important contributions are discussed in Section 3.1. Section 3.2 presents works in the field of sound synthesis, i.e., techniques to simulate physically plausible sounds in a virtual environment. In Section 3.3 we introduce works that analyze instruments to establish a connection between sound and shape.

Chapter 4 describes the steps in our natural frequency optimization pipeline and represents the primary contribution of this thesis. We first introduce three shape parametrization methods for the design space of our shape optimization framework. The section on finite element models develops the element types we explored and discusses their implementation. Then we describe our formulation of the frequency optimization problem in order to find feasible solutions in the design spaces.

Chapter 5 presents results obtained with the shape optimization framework and discusses our methods of evaluating the quality of the solutions. We describe our workflow to measure the natural frequencies in the audible spectrum of a given physical object and compare the computational solutions of our finite element framework to measurements using the example of aluminium plates. In order to increase the accuracy of the frequency predictions, we perform material parameter estimation on the same test objects. In this chapter, we also document our workflow for manufacturing our own idiophones from tin and compare the finite element predictions to the experimentally determined frequencies. Finally, we present a result that demonstrates the optimization of harmonic ratios in our design space.

Chapter 6 discusses a possibility to speed up frequency optimization by using coarsely resolved meshes. To this end, we evaluate how mesh resolution and the presence of fine surface details influence the natural frequency predictions for an object.

Chapter 7 summarizes the major findings of this thesis and gives an outlook on future research directions.

Theoretical Foundations

2.1 Linear Elasticity Theory

Elasticity theory is the subfield of continuum mechanics that describes the motions and deformations of solid objects. Solid objects are modeled as continua that respond to external forces with internal stresses and that deform in response to these stresses. If deformations are expected to be small, the relationships between all occurring quantities can be linearized, which significantly simplifies computation. The basics of elasticity theory are discussed here because they lead to the development of equations of motions that model the behavior of solids. We are ultimately interested in the description of vibrations that lie in the audible spectrum, i.e., have a frequency between 20 and 20,000 Hz. These vibrations introduce only small deformations to the object and are therefore described accurately by the linear theory.

The theory of elasticity is usually presented in three parts: kinematics, kinetics, and constitutive relations. In combination, they culminate in the equations of motion for linear elasticity, which yield a full description of solid object behavior.

Kinematics is purely descriptive and characterizes the motions and deformations of an object in terms of a mathematical language. The most important assumptions are that a solid object is a continuum, i.e., the atomic structure is ignored, and that deformations are small enough to ignore quadratic terms. Kinetics is concerned with the causality between physical forces and the resulting acceleration of the object. This involves body forces such as gravity, surface forces such as collisions with other objects, as well as internal forces that develop due to deformations. Constitutive relations describe the properties of the material as idealized stress-strain laws. While the results of kinematics and kinetics are derived from fundamental principles, constitutive laws are assumptions that have proven to hold well in practice.

In the remainder of this section, we present the most important results of linear elasticity theory. More thorough treatments of this subject can be found in engineering textbooks [TG69, MEH12, Fel13].

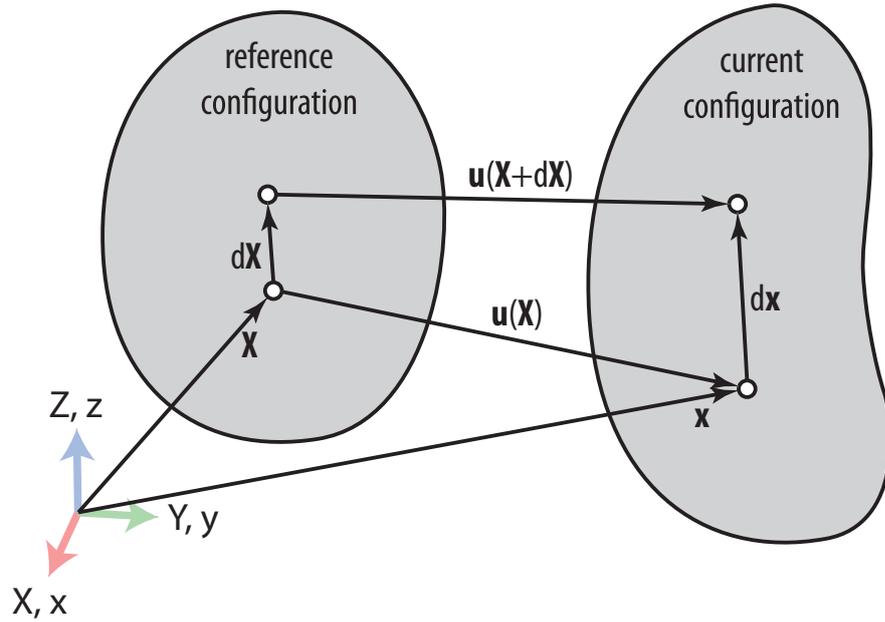


Figure 2.1: Reference configuration and current configuration of an elastic body. The points \mathbf{X} and its infinitesimal neighbor $\mathbf{X} + d\mathbf{X}$ of the reference configuration are displaced to the points \mathbf{x} and $\mathbf{x} + d\mathbf{x}$ in the current configuration.

2.1.1 Kinematics

In the beginning, we introduce notation to describe a solid object as a moving and deforming continuum. We define the reference configuration of a solid as a compact and connected set $\Omega \subset \mathbb{R}^3$. This configuration is the starting situation and describes the state of the object in the absence of forces. The position of a particle in its reference configuration is denoted as $\mathbf{X} \in \mathbb{R}^3$, see Fig. 2.1.

The solid may deform over time, and this way a particle can move from its reference configuration to the current configuration. Its position in the current configuration is denoted as $\mathbf{x} \in \mathbb{R}^3$. If we want to emphasize the dependence of \mathbf{x} on its corresponding reference point \mathbf{X} and on time t , we denote it as $\mathbf{x}(\mathbf{X}, t)$. The displacement between the current and the reference configuration is denoted as $\mathbf{u}(\mathbf{X}, t) = \mathbf{x}(\mathbf{X}, t) - \mathbf{X}$. This is an important variable because our ultimate goal will be to find \mathbf{u} as a function of time because this describes the behavior of an object in its entirety.

The next quantity we introduce describes the deformation in the neighborhood of a point \mathbf{X} as the solid moves from the reference configuration to the current configuration. To this end we define a point $\mathbf{X} + d\mathbf{X}$ that is infinitesimally close to \mathbf{X} . These two points map to \mathbf{x} and $\mathbf{x} + d\mathbf{x}$ in the current configuration. As a quantification of deformation, we look at the difference in length between the line segment connecting \mathbf{X} and $\mathbf{X} + d\mathbf{X}$ and its image in the current configuration. To facilitate analysis, we measure the difference between the squared lengths $dS^2 = d\mathbf{X}^T d\mathbf{X}$ and $ds^2 = d\mathbf{x}^T d\mathbf{x}$.

A look at Fig. 2.1 reveals that

$$d\mathbf{x} = d\mathbf{X} + \mathbf{u}(\mathbf{X} + d\mathbf{X}) - \mathbf{u}(\mathbf{X}). \quad (2.1)$$

The central term can be developed into a Taylor series

$$\mathbf{u}(\mathbf{X} + d\mathbf{X}) = \mathbf{u}(\mathbf{X}) + \frac{\partial \mathbf{u}}{\partial \mathbf{X}} d\mathbf{X}, \quad (2.2)$$

which can be truncated after the linear term because $d\mathbf{X}$ is an infinitesimal quantity. Note that

$$\nabla \mathbf{u} = \frac{\partial \mathbf{u}}{\partial \mathbf{X}} = \begin{pmatrix} \frac{\partial u_1}{\partial X_1} & \frac{\partial u_1}{\partial X_2} & \frac{\partial u_1}{\partial X_3} \\ \frac{\partial u_2}{\partial X_1} & \frac{\partial u_2}{\partial X_2} & \frac{\partial u_2}{\partial X_3} \\ \frac{\partial u_3}{\partial X_1} & \frac{\partial u_3}{\partial X_2} & \frac{\partial u_3}{\partial X_3} \end{pmatrix}$$

is the Jacobi matrix of \mathbf{u} with respect to \mathbf{X} . Substitution of Eq. (2.2) into Eq. (2.1) yields

$$d\mathbf{x} = d\mathbf{X} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}} d\mathbf{X}.$$

If we plug this formula into our deformation measure, we obtain

$$\begin{aligned} ds^2 - dS^2 &= \left(d\mathbf{X} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}} d\mathbf{X} \right)^T \left(d\mathbf{X} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}} d\mathbf{X} \right) - d\mathbf{X}^T d\mathbf{X} = \\ &= d\mathbf{X}^T \left[\frac{\partial \mathbf{u}}{\partial \mathbf{X}} + \left(\frac{\partial \mathbf{u}}{\partial \mathbf{X}} \right)^T + \left(\frac{\partial \mathbf{u}}{\partial \mathbf{X}} \right)^T \frac{\partial \mathbf{u}}{\partial \mathbf{X}} \right] d\mathbf{X}, \end{aligned}$$

which is usually written as

$$ds^2 - dS^2 = d\mathbf{X}^T \cdot 2\mathbf{E} \cdot d\mathbf{X}.$$

The symmetric second-order tensor \mathbf{E} is referred to as the Green-Lagrange strain tensor. It is given by

$$\mathbf{E} = \frac{1}{2} \left[(\nabla \mathbf{u})^T + \nabla \mathbf{u} + (\nabla \mathbf{u})^T \nabla \mathbf{u} \right].$$

If the deformation gradients $\nabla \mathbf{u}$ are small, the quadratic term in \mathbf{E} can be neglected. The resulting second-order tensor

$$\boldsymbol{\varepsilon} = \frac{1}{2} \left[(\nabla \mathbf{u})^T + \nabla \mathbf{u} \right] \quad (2.3)$$

is referred to as the infinitesimal strain tensor or, in short, strain. Because it is symmetric by definition, it has only six independent variables

$$\boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ & \varepsilon_{22} & \varepsilon_{23} \\ \text{symm.} & & \varepsilon_{33} \end{pmatrix}.$$

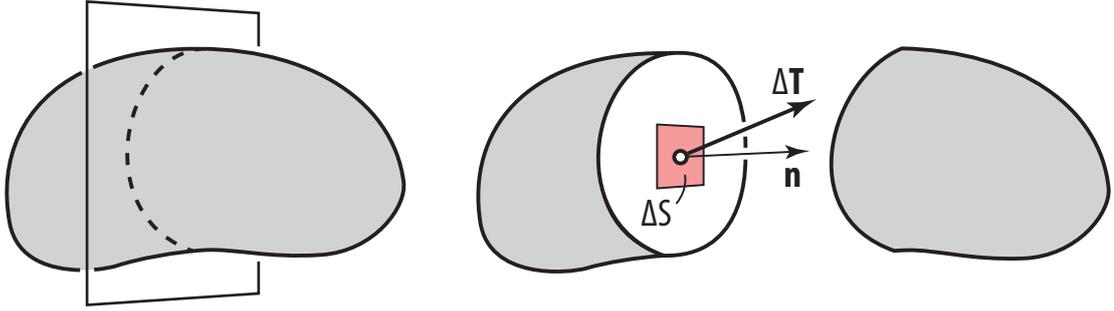


Figure 2.2: A mental planar cut is made through an elastic body. The mean stress is defined as the ratio between the force acting on a surface element and its area.

The transition from \mathbf{E} to $\boldsymbol{\varepsilon}$ represents the eponymous linearization of linear elasticity theory. The diagonal elements of $\boldsymbol{\varepsilon}$ are known as normal deformations, and the off-diagonal elements as shear deformations. These names refer to the type of geometric transformation that the corresponding deformations induce if present.

The six equations contained in Eq. (2.3), which connect the six strain components and the three displacement components, are referred to as the kinematic equations and are part of the equations of motion of linear elasticity theory.

2.1.2 Kinetics

This section explains how the forces acting on and inside of a solid relate to the acceleration of its constitutive points. The central concept is that of stress, which describes the forces acting on a surface. To define the stress at a point \mathbf{x} in the interior of the object domain, we make a mental planar cut at that point, thereby splitting Ω in two parts, see Fig. 2.2. Let ΔS be the area of a surface element around \mathbf{x} , $\Delta \mathbf{T}$ the force acting on said surface element due to internal force interactions, and \mathbf{n} the normal of the surface element. Then

$$\mathbf{t}_m^{(\mathbf{n})} = \frac{\Delta \mathbf{T}}{\Delta S}$$

defines the mean stress vector. The vector $\Delta \mathbf{T}$ has the units of force, but stress describes force per unit area. By shrinking the surface element S and taking the limit

$$\mathbf{t}^{(\mathbf{n})} = \lim_{\Delta S \rightarrow 0} \frac{\Delta \mathbf{T}}{\Delta S}, \quad (2.4)$$

we arrive at the stress vector $\mathbf{t}^{(\mathbf{n})}$ in \mathbf{x} with respect to the normal \mathbf{n} of the surface element. Note that the corresponding stress vector in \mathbf{x} on the other side of the cut is $\mathbf{t}^{(-\mathbf{n})}$ because the two surface elements close seamlessly at \mathbf{x} . Furthermore, it holds that $-\mathbf{t}^{(\mathbf{n})} = \mathbf{t}^{(-\mathbf{n})}$ due to Newton's third law of action and reaction.

To describe the stress state at a point \mathbf{x} in the current configuration, the Cauchy stress tensor $\boldsymbol{\sigma}$ is introduced. We define a cuboid with infinitesimal side lengths dx_1 , dx_2 , and dx_3 that is anchored at \mathbf{x} and aligned with the orthonormal basis $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$. The

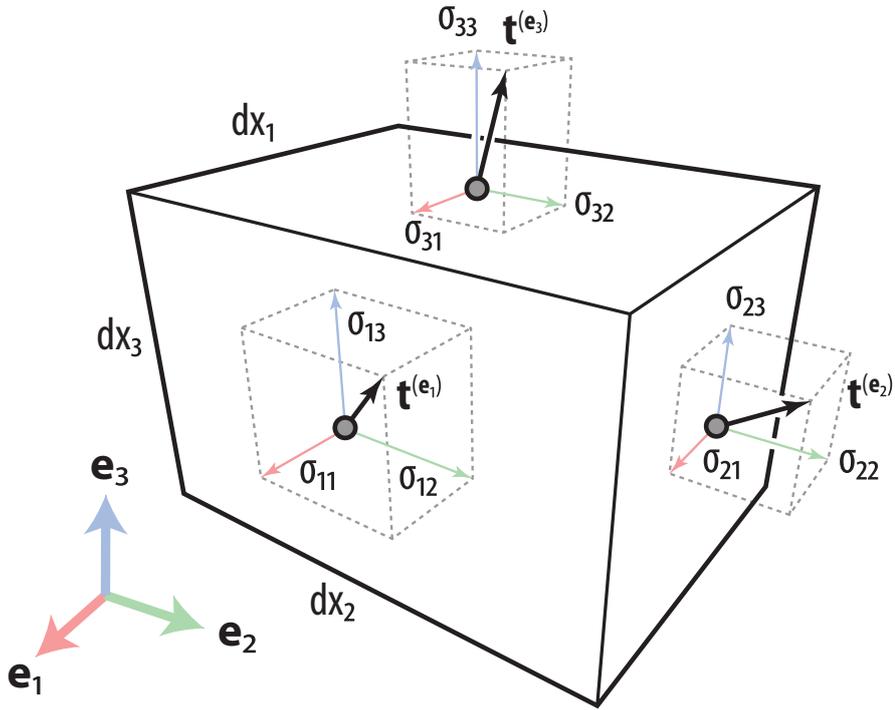


Figure 2.3: The stress vectors acting on the positively-oriented faces of a cuboid define the coefficients of the Cauchy stress tensor.

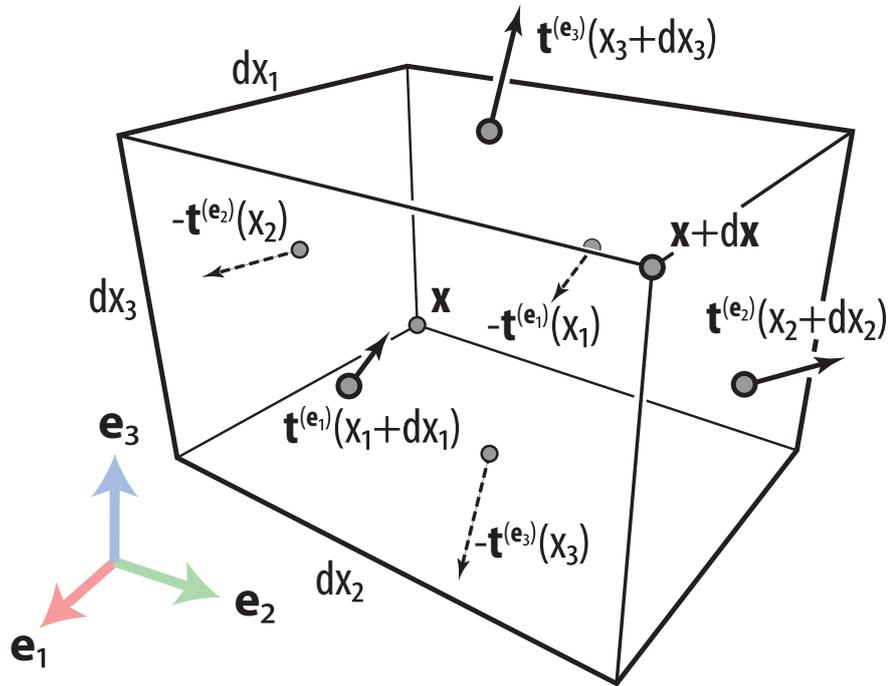


Figure 2.4: The stress vectors acting on all faces of the cuboid.

stress vectors on the positively-oriented faces of the cuboid, i.e., the faces whose normals equal the basis vectors, are obtained by shrinking the cuboid to a point and taking the limit as in Eq. (2.4). This yields

$$\begin{aligned}\mathbf{t}^{(\mathbf{e}_1)} &= \sigma_{11}\mathbf{e}_1 + \sigma_{12}\mathbf{e}_2 + \sigma_{13}\mathbf{e}_3, \\ \mathbf{t}^{(\mathbf{e}_2)} &= \sigma_{21}\mathbf{e}_1 + \sigma_{22}\mathbf{e}_2 + \sigma_{23}\mathbf{e}_3, \\ \mathbf{t}^{(\mathbf{e}_3)} &= \sigma_{31}\mathbf{e}_1 + \sigma_{32}\mathbf{e}_2 + \sigma_{33}\mathbf{e}_3.\end{aligned}$$

As illustrated in Fig. 2.3, the coefficients σ_{ij} constitute the Cauchy stress tensor

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix}.$$

The diagonal elements are known as normal stresses, and the off-diagonal elements as shear stresses.

In order to prove the symmetry of the Cauchy stress tensor, we look at the stress vectors acting on all six faces of the cuboid, as illustrated in Fig. 2.4. By taking moments about the basis vector \mathbf{e}_1 , we obtain

$$\begin{aligned}& \sigma_{23}(x_2 + dx_2) dx_1 dx_3 \frac{dx_2}{2} + \sigma_{23}(x_2) dx_1 dx_3 \frac{dx_2}{2} \\ & - \sigma_{32}(x_3 + dx_3) dx_1 dx_2 \frac{dx_3}{2} - \sigma_{32}(x_3) dx_1 dx_2 \frac{dx_3}{2} = 0.\end{aligned}$$

In this equation, $\sigma_{23}(x_2 + dx_2) dx_1 dx_3$ is the force acting on the face with normal \mathbf{e}_2 and $\frac{dx_2}{2}$ is its distance to \mathbf{x} . By shrinking the cuboid to a point, we take the limits $dx_2 \rightarrow 0$ and $dx_3 \rightarrow 0$, which yields $\sigma_{23} = \sigma_{32}$. Taking moments about \mathbf{e}_2 and \mathbf{e}_3 as well proves the symmetry of the Cauchy stress tensor

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ & \sigma_{22} & \sigma_{23} \\ \text{symm.} & & \sigma_{33} \end{pmatrix}.$$

Equipped with this definition, Newton's second law

$$d\mathbf{R} = \mathbf{b} dm, \tag{2.5}$$

where $d\mathbf{R}$ is the resultant of forces, dm is the mass, and \mathbf{b} is the acceleration of an infinitesimal volume element, can be specialized to a deformable solid.

Again, we start with an infinitesimally small cuboid element that is aligned with the coordinate system $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ and anchored at \mathbf{x} . The stress vectors acting on the six cuboid faces are

$$\begin{aligned}-\mathbf{t}^{(\mathbf{e}_1)}(x_1), & \quad \mathbf{t}^{(\mathbf{e}_1)}(x_1 + dx_1), \\ -\mathbf{t}^{(\mathbf{e}_2)}(x_2), & \quad \mathbf{t}^{(\mathbf{e}_2)}(x_2 + dx_2), \\ -\mathbf{t}^{(\mathbf{e}_3)}(x_3), & \quad \mathbf{t}^{(\mathbf{e}_3)}(x_3 + dx_3).\end{aligned}$$

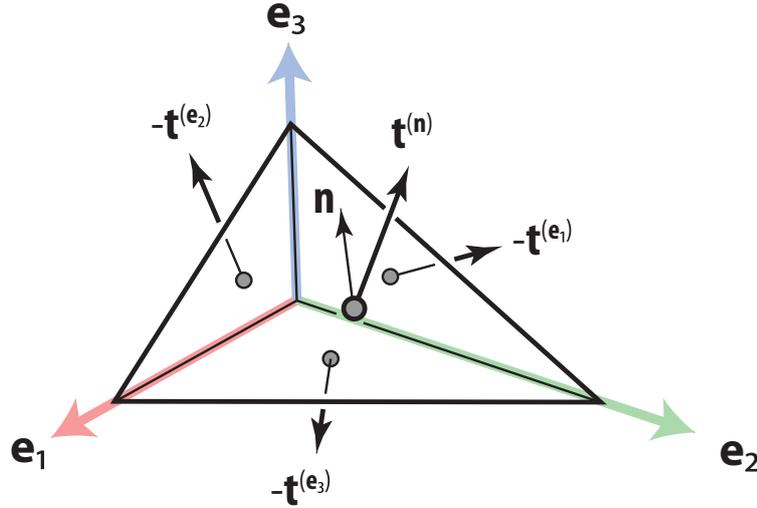


Figure 2.5: An infinitesimal tetrahedral volume element.

The stress vectors acting on the positively oriented faces can be expanded into the Taylor series

$$\mathbf{t}^{(e_i)}(x_i + dx_i) = \mathbf{t}^{(e_i)}(x_i) + \frac{\partial \mathbf{t}^{(e_i)}(x_i)}{\partial x_i} dx_i + \dots$$

The resultant of forces $d\mathbf{R}$ acting on the cuboid element is given by the sum of body forces $\mathbf{f}dV$ and the stresses acting on the cuboid faces. Here, \mathbf{f} denotes volume force density, and dV is the volume of the cuboid. By using the Taylor expansions, we arrive at

$$d\mathbf{R} = \left[\mathbf{f} + \frac{\partial \mathbf{t}^{(e_1)}(x_1)}{\partial x_1} + \frac{\partial \mathbf{t}^{(e_2)}(x_2)}{\partial x_2} + \frac{\partial \mathbf{t}^{(e_3)}(x_3)}{\partial x_3} \right] dV + \dots,$$

where $dx_1 dx_2 dx_3$ has been replaced with dV . Division by dV and setting $dm = \rho dV$, where ρ is the density of the solid, yields

$$\frac{\partial \mathbf{t}^{(e_1)}(x_1)}{\partial x_1} + \frac{\partial \mathbf{t}^{(e_2)}(x_2)}{\partial x_2} + \frac{\partial \mathbf{t}^{(e_3)}(x_3)}{\partial x_3} + \mathbf{f} = \rho \mathbf{b}.$$

Using the divergence operator and the Cauchy stress tensor, this can be written more concisely as

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{f} = \rho \mathbf{b}. \quad (2.6)$$

This three-component equation is known as the fundamental law of dynamics and is part of the equations of motion of a solid deformable continuum.

For the derivation of the principle of virtual work in Section 2.2, we need one more result known as Cauchy's stress theorem. It states that, by knowing the Cauchy stress tensor at a point \mathbf{x} , we can determine the stress vector for an arbitrary surface normal \mathbf{n} as

$$\mathbf{t}^{(\mathbf{n})} = \boldsymbol{\sigma} \cdot \mathbf{n}. \quad (2.7)$$

In order to prove this, we imagine an infinitesimally small tetrahedral volume element, three of whose faces are aligned with the basis vectors \mathbf{e}_1 , \mathbf{e}_2 , and \mathbf{e}_3 , as seen in Fig. 2.5. Let $\mathbf{n} = (n_1, n_2, n_3)^T$ denote the normal of the fourth face of the tetrahedron. If $dS^{(\mathbf{n})}$ denotes the surface area of this face, the surface areas of the other faces are given by the projections of \mathbf{n} onto the basis vectors. This yields the expressions

$$\begin{aligned} dS^{(\mathbf{e}_1)} &= n_1 dS^{(\mathbf{n})}, \\ dS^{(\mathbf{e}_2)} &= n_2 dS^{(\mathbf{n})}, \\ dS^{(\mathbf{e}_3)} &= n_3 dS^{(\mathbf{n})}. \end{aligned}$$

Specializing Newton's second law Eq. (2.5) to the tetrahedral element yields

$$\mathbf{f} dV - \mathbf{t}^{(\mathbf{e}_1)} n_1 dS^{(\mathbf{n})} - \mathbf{t}^{(\mathbf{e}_2)} n_2 dS^{(\mathbf{n})} - \mathbf{t}^{(\mathbf{e}_3)} n_3 dS^{(\mathbf{n})} + \mathbf{t}^{(\mathbf{n})} dS^{(\mathbf{n})} = \mathbf{b}^{(\mathbf{n})} \rho dV,$$

where \mathbf{f} denotes volume force density and dV the volume of the element. Substituting $dV = \frac{1}{3} dh dS^{(\mathbf{n})}$ and dividing by $dS^{(\mathbf{n})}$ leads to

$$\mathbf{f} \frac{1}{3} dh - \mathbf{t}^{(\mathbf{e}_1)} n_1 - \mathbf{t}^{(\mathbf{e}_2)} n_2 - \mathbf{t}^{(\mathbf{e}_3)} n_3 + \mathbf{t}^{(\mathbf{n})} = \mathbf{b} \rho \frac{1}{3} dh.$$

If we shrink the tetrahedron to point and take the limit $dh \rightarrow 0$, we finally obtain

$$\mathbf{t}^{(\mathbf{n})} = \mathbf{t}^{(\mathbf{e}_1)} n_1 + \mathbf{t}^{(\mathbf{e}_2)} n_2 + \mathbf{t}^{(\mathbf{e}_3)} n_3 = \boldsymbol{\sigma} \cdot \mathbf{n},$$

which proves Cauchy's stress theorem.

2.1.3 Constitutive Equations

So far we have introduced 15 unknowns: the three components of the displacement \mathbf{u} , the six independent components of the Green-Lagrange strain tensor $\boldsymbol{\varepsilon}$, and the six independent components of the Cauchy stress tensor $\boldsymbol{\sigma}$. With the six equations from Eqs. (2.3), and the three equations from Eqs. (2.6), we are still six equations short of a full set of equations of motion.

Constitutive equations provide these additional constraints by describing the relationship between strains and stresses. They can be experimentally determined for specific materials using universal testing machines. The general form of constitutive equations in linear elasticity is

$$\boldsymbol{\varepsilon} = \mathcal{F}(\boldsymbol{\sigma}).$$

For many materials, the relationship between stress and strain is linear for small deformations and can be written as

$$\boldsymbol{\sigma} = \mathbf{E} \boldsymbol{\varepsilon}, \tag{2.8}$$

which is known as the generalized Hooke's Law. The fourth-order tensor \mathbf{E} is referred to as the stiffness tensor. A fourth-order tensor consists of up to 81 independent scalar variables. However, due to the symmetry of $\boldsymbol{\sigma}$, $\boldsymbol{\varepsilon}$, and \mathbf{E} itself, \mathbf{E} has at most 21 independent scalars.

To express Eq. 2.8 in matrix notation, we write $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ as vectors and obtain

$$\begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{23} \\ \sigma_{31} \end{pmatrix} = \begin{pmatrix} E_{11} & E_{12} & E_{13} & E_{14} & E_{15} & E_{16} \\ & E_{22} & E_{23} & E_{24} & E_{25} & E_{26} \\ & & E_{33} & E_{34} & E_{35} & E_{36} \\ & & & E_{44} & E_{45} & E_{46} \\ & & & & E_{55} & E_{56} \\ \text{symm.} & & & & & E_{66} \end{pmatrix} \begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{12} \\ 2\varepsilon_{23} \\ 2\varepsilon_{31} \end{pmatrix}.$$

For many material, further simplifications can be made. A material is called orthotropic if

1. there are three mutually orthogonal planes of symmetry,
2. shear stresses along these planes do not affect normal deformations,
3. normal stresses along these planes do not affect shear deformations, and
4. shear stresses only affect the corresponding shear deformations.

In this case, the nine values in the upper-right quadrant (and hence the lower-left quadrant) of \mathbf{E} equal zero because of conditions 2 and 3. Additionally, the off-diagonal elements in the lower-right quadrant equal zero because of condition 4.

Even more simplifications can be made for isotropic materials, i.e., materials whose behavior is completely direction-independent. In this case, two parameters suffice to determine the remaining components of \mathbf{E} . The first parameter is Young's Modulus E , which describes the ratio between stress and strain in a uniaxial tensile test: $\sigma = E\varepsilon$. The second parameter is Poisson's ratio ν , which relates the prolongation of a prismatic member in tension to its decrease in diameter. Hooke's Law for isotropic materials can be written as

$$\begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{23} \\ \sigma_{31} \end{pmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1-\nu & \nu & \nu & & & \\ \nu & 1-\nu & \nu & & & \\ \nu & \nu & 1-\nu & & & \\ & & & \frac{1}{2}-\nu & & \\ & & & & \frac{1}{2}-\nu & \\ & & & & & \frac{1}{2}-\nu \end{pmatrix} \begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{12} \\ 2\varepsilon_{23} \\ 2\varepsilon_{31} \end{pmatrix}. \quad (2.9)$$

For the rest of this thesis, we will use this constitutive law because metals are isotropic.

The kinematic equations (2.3), the fundamental law of dynamics (2.6), and the constitutive equations (2.9) add up to 15 equations for 15 variables that describe the motions of a deformable solid in full:

$$\begin{aligned} \boldsymbol{\varepsilon} &= \frac{1}{2} [(\nabla \mathbf{u})^T + \nabla \mathbf{u}], \\ \boldsymbol{\sigma} &= \mathbf{E} \boldsymbol{\varepsilon}, \\ \nabla \cdot \boldsymbol{\sigma} + \mathbf{f} &= \rho \mathbf{b}. \end{aligned} \quad (2.10)$$

2.2 Virtual Work and Energy Principles

The principle of virtual work is an alternative but equivalent formulation of the motions of deformable solids. The equations derived in Section 2.1 directly describe the behavior of each point in the continuum. In contrast to this, the principle of virtual work is a global formulation that corresponds to the principle of stationary action in classical mechanics. We review it here because it serves as the basis for the development of the finite element method in solid mechanics.

Concisely stated, the principle of virtual work says the following: If the current configuration of the deformable body is frozen in time, an arbitrary infinitesimal, admissible displacement does zero mechanical work. This displacement is referred to as “virtual” because it is not a displacement that the body actually undergoes and is denoted as $\delta \mathbf{u}$. The work that the body does as a result of the virtual displacement is called virtual work and is denoted as δA . The virtual displacement has to respect the essential boundary conditions of the body, e.g., $\delta \mathbf{u} = 0$ at a point that has a fixed support.

The virtual work done by the surface stress vectors $\mathbf{t}^{(n)}$ due to an arbitrary virtual displacement $\delta \mathbf{u}$ on the boundary $\partial\Omega$ of Ω is

$$\int_{\partial\Omega} [\mathbf{t}^{(n)}]^T \delta \mathbf{u} dA.$$

Using Cauchy’s stress theorem Eq. (2.7) and the divergence theorem, we can transform this surface integral to a volume integral and obtain

$$\begin{aligned} \int_{\partial\Omega} [\mathbf{t}^{(n)}]^T \delta \mathbf{u} dA &= \int_{\partial\Omega} (\delta \mathbf{u})^T \boldsymbol{\sigma} \mathbf{n} dA = \int_{\Omega} \nabla \cdot (\boldsymbol{\sigma} \delta \mathbf{u}) dV = \\ &= \int_{\Omega} (\nabla \cdot \boldsymbol{\sigma})^T \delta \mathbf{u} dV + \int_{\Omega} \boldsymbol{\sigma}^T : \nabla \delta \mathbf{u} dV, \end{aligned} \quad (2.11)$$

where the operator “:” denotes the Frobenius product, i.e., the component-wise inner product of two matrices, also referred to as double tensor contraction. Using the fundamental law of dynamics Eq. (2.6), we can rewrite the penultimate integral as

$$\int_{\Omega} (\nabla \cdot \boldsymbol{\sigma})^T \delta \mathbf{u} dV = \int_{\Omega} (-\mathbf{f} + \rho \mathbf{b})^T \delta \mathbf{u} dV. \quad (2.12)$$

Using $A : B = A^T : B^T$ and considering that $\boldsymbol{\sigma}$ is symmetric, we can relate the ultimate integral to the infinitesimal strain tensor by

$$\int_{\Omega} \boldsymbol{\sigma}^T : \nabla \delta \mathbf{u} dV = \int_{\Omega} \boldsymbol{\sigma} : \frac{1}{2} [\nabla \delta \mathbf{u} + (\nabla \delta \mathbf{u})^T] dV = \int_{\Omega} \boldsymbol{\sigma} : \delta \boldsymbol{\varepsilon} dV. \quad (2.13)$$

Substituting Eqs. (2.12) and (2.13) back into Eq. (2.11) and reordering the terms yields the principle of virtual work

$$\underbrace{- \int_{\Omega} \boldsymbol{\sigma} : \delta \boldsymbol{\varepsilon} dV}_{\delta A_{\text{in}}} + \underbrace{\int_{\Omega} \mathbf{f}^T \delta \mathbf{u} dV + \int_{\partial\Omega} [\mathbf{t}^{(n)}]^T \delta \mathbf{u} dA}_{\delta A_{\text{ex}}} - \underbrace{\int_{\Omega} \rho \mathbf{b}^T \delta \mathbf{u} dV}_{\delta A_{\text{fi}}} = 0. \quad (2.14)$$

This states that the sum of internal virtual work δA_{in} , external virtual work δA_{ex} , and virtual work done by inertial forces δA_{fi} vanishes. Inertial forces, also known as fictitious forces or d'Alembert forces, are introduced such that the dynamics problem at hand can be formulated as a statics problem. This technique is known as d'Alembert's principle. In the language of Lagrangian mechanics, the principle of virtual work states that among all paths of motion that a deformable solid can follow, nature chooses the one that makes Eq. (2.14) true.

The basis for finite element models in solid mechanics is an energy formulation of the principle of virtual work. We do not discuss the energy equivalent of external virtual work here because it will not play a part in our application. Instead, we focus on the energy equivalents of internal virtual work—strain energy—and virtual work done by inertial forces—kinetic energy.

For the remainder of this section, we use the symbols $\boldsymbol{\varepsilon}$ and $\boldsymbol{\sigma}$ to denote the linearized versions of the tensors, such that we can write tensor contractions as matrix-vector multiplications. The strain energy density for a linear constitutive relationship such as Eq. (2.8) can then be defined as

$$\mathcal{U}_0 = \frac{1}{2} \boldsymbol{\varepsilon}^T \mathbf{E} \boldsymbol{\varepsilon} = \frac{1}{2} \boldsymbol{\sigma}^T \boldsymbol{\varepsilon}.$$

It follows immediately that

$$\boldsymbol{\sigma}^T = \frac{\partial \mathcal{U}_0}{\partial \boldsymbol{\varepsilon}}.$$

If we rewrite the definition of internal virtual work δA_{in} in terms of differentials instead of variations, we can make the substitutions

$$dA_{\text{in}} = - \int_{\Omega} \boldsymbol{\sigma}^T d\boldsymbol{\varepsilon} dV = - \int_{\Omega} \frac{\partial \mathcal{U}_0}{\partial \boldsymbol{\varepsilon}} d\boldsymbol{\varepsilon} dV = - \int_{\Omega} d\mathcal{U}_0 dV.$$

The last integral

$$d\mathcal{U} = \int_{\Omega} d\mathcal{U}_0 dV$$

is known as strain energy.

A similar transformation is possible for the virtual work done by inertial forces. In this case, we use the definition of kinetic energy density

$$\mathcal{T}_0 = \frac{1}{2} \rho \dot{\mathbf{u}}^T \dot{\mathbf{u}}$$

and verify that

$$\rho \dot{\mathbf{u}}^T = \frac{\partial \mathcal{T}_0}{\partial \dot{\mathbf{u}}}.$$

Using the relationship

$$\mathbf{b}^T d\mathbf{u} = \left(\frac{\partial \dot{\mathbf{u}}}{\partial t} \right)^T d\mathbf{u} = \left(\frac{\partial \dot{\mathbf{u}}}{\partial t} \right)^T \frac{\partial \mathbf{u}}{\partial t} dt = \left(\frac{\partial \mathbf{u}}{\partial t} \right)^T \frac{\partial \dot{\mathbf{u}}}{\partial t} dt = \dot{\mathbf{u}}^T d\dot{\mathbf{u}}, \quad (2.15)$$

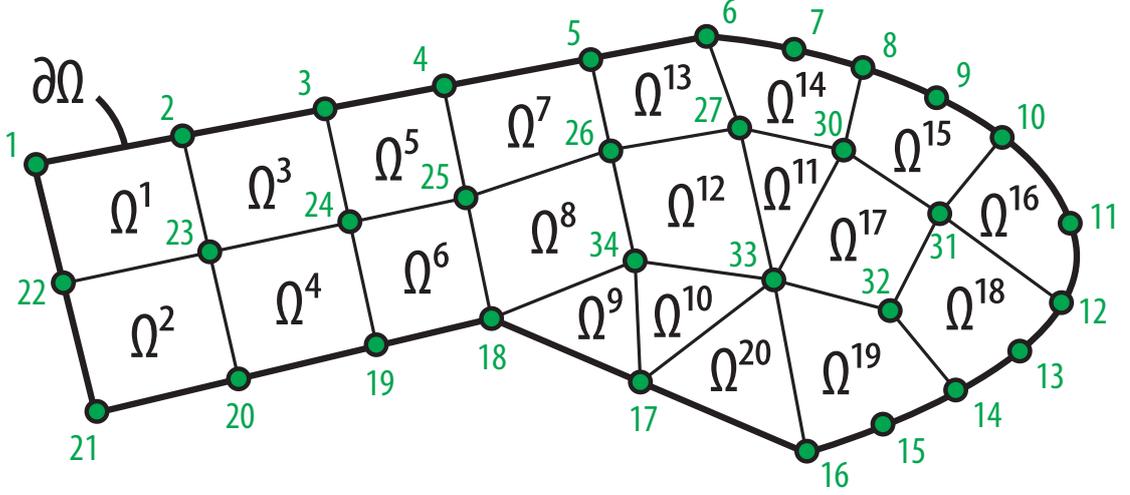


Figure 2.6: Example of a finite element mesh. The domain Ω is partitioned into 20 elements $\Omega^1, \dots, \Omega^{20}$. There are triangular, quadrilateral, and curved-boundary elements. The element domains are defined by a total of 34 nodes.

we can also rewrite the virtual work done by inertial forces in terms of energy:

$$dA_{\text{fi}} = - \int_{\Omega} \rho \mathbf{b}^T d\mathbf{u} dV = - \int_{\Omega} \rho \dot{\mathbf{u}}^T d\dot{\mathbf{u}} dV = - \int_{\Omega} \frac{\partial \mathcal{T}_0}{\partial \dot{\mathbf{u}}} d\dot{\mathbf{u}} dV = - \int_{\Omega} d\mathcal{T}_0 dV.$$

The last integral

$$d\mathcal{T} = \int_{\Omega} d\mathcal{T}_0 dV$$

is known as kinetic energy.

As mentioned before, we ignore the external virtual work δA_{ex} for the purposes of this thesis. This corresponds to the fact that neither body forces \mathbf{f} nor prescribed surface stresses $\mathbf{t}^{(\mathbf{n})}$ will occur in our models. Therefore, external virtual work always equals zero.

With all energy definitions in place, the principle of virtual work Eq. (2.14) can be written in variational form as

$$\delta(\mathcal{U} + \mathcal{T}) = 0, \quad (2.16)$$

which is a principle of stationary energy.

2.3 Finite Element Discretization

The stationary energy principle Eq. (2.16) is a compact description of the behavior of a deformable solid. However, due its integral nature and infinite number of degrees of freedom (DoF), it has a closed-form solution for but the simplest of examples.

The goal of the finite element method for solid mechanics is to provide a formulation that satisfies Eq. (2.16) in an approximate sense while reducing the DoF to a finite

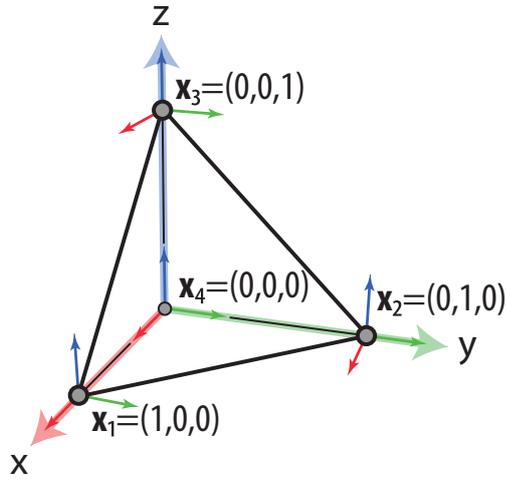


Figure 2.7: The standard linear tetrahedron with three degrees of freedom at each node.

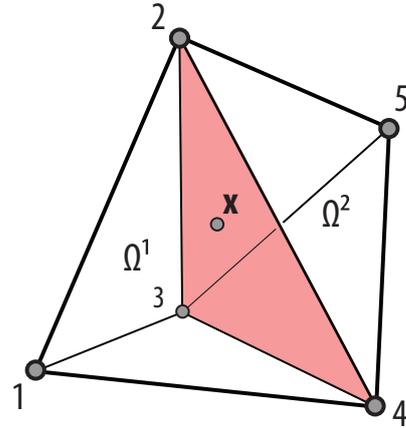


Figure 2.8: Two tetrahedral elements are joined at a face, sharing three nodes. A quantity must be interpolated the same way throughout the face by both elements.

number. The primary variable that represents these DoF is the displacement \mathbf{u} at a finite number of points throughout the domain Ω of the solid. Successful application of the finite element method replaces Eq. (2.16) with a linear system in \mathbf{u} .

The central discretization concept of finite elements is the partitioning of Ω into a set of elements Ω^e . There is a wide variety of element shapes, from linear tetrahedra, bricks, and triangular prisms to elements with curved boundaries. Every element type has a fixed number of nodes, which define its number of DoF. A 2d example is given in Fig. 2.6, where a domain with a curved boundary is partitioned into a set of triangles, quads, and curved elements at the boundary.

The basics of finite element analysis will be explained by working out a toy example: the standard linear tetrahedral element spanned by the vertices

$$\begin{aligned}\mathbf{x}_1 &= (1, 0, 0)^T, \\ \mathbf{x}_2 &= (0, 1, 0)^T, \\ \mathbf{x}_3 &= (0, 0, 1)^T, \\ \mathbf{x}_4 &= (0, 0, 0)^T.\end{aligned}$$

The linear tetrahedral element is the simplest 3d element and has one node at each corner. A displacement vector $\mathbf{u}_i \in \mathbb{R}^3$, $i \in \{1, \dots, 4\}$ is defined at each node, totaling twelve DoF, see Fig. 2.7. On the edges, faces and in the interior of the tetrahedron, the displacements are interpolated from the nodes by using barycentric coordinates. Thus, the displacements vary linearly across this element. Other quantities like the strain $\boldsymbol{\varepsilon}$

can be inferred by applying the definition Eq. (2.3) to the interpolated displacements instead of the (unknown) real displacements.

To interpolate any quantity q at a point $\mathbf{x} = (x_1, x_2, x_3)^T$ in the standard tetrahedron, we can use

$$\bar{q}(\mathbf{x}) = x_1 q_1 + x_2 q_2 + x_3 q_3 + (1 - x_1 - x_2 - x_3) q_4,$$

where the q_i represent the values of the quantity q at the 4 corner nodes. For example, applying this formula at the edge midpoint $(0.5, 0, 0)^T$ between nodes 1 and 4 yields the arithmetic mean of q_1 and q_4 . The coefficients of q_i can be interpreted as the influence of a node on the interior of the element and are referred to as shape functions N_i^e :

$$\begin{aligned} N_1^e(\mathbf{x}) &= x_1, \\ N_2^e(\mathbf{x}) &= x_2, \\ N_3^e(\mathbf{x}) &= x_3, \\ N_4^e(\mathbf{x}) &= 1 - x_1 - x_2 - x_3. \end{aligned}$$

The shape functions meet a number of conditions that are necessary to qualify them as a proper way of interpolation. They are a partition of unity on the element domain, i.e.,

$$\sum N_i^e(\mathbf{x}) = 1, \quad \forall \mathbf{x} \in \Omega^e.$$

Also, it holds that

$$\begin{aligned} N_i^e(\mathbf{x}_i) &= 1, \\ N_i^e(\mathbf{x}_j) &= 0, \quad i \neq j \end{aligned}$$

to ensure that the nodal value q_i is attained through interpolation at point \mathbf{x}_i .

Additionally, the shape functions are continuous across element borders. For example, if two elements share three nodes because they are joined at a face, the interpolation formula yields the same value for points on the face, no matter on which one of the two elements it is evaluated. Fig. 2.8 illustrates this situation with two tetrahedral elements Ω^1 and Ω^2 . For the point \mathbf{x} it must hold that

$$\begin{aligned} N_1^1(\mathbf{x})q_1 + N_2^1(\mathbf{x})q_2 + N_3^1(\mathbf{x})q_3 + N_4^1(\mathbf{x})q_4 &= \\ = N_2^2(\mathbf{x})q_2 + N_3^2(\mathbf{x})q_3 + N_4^2(\mathbf{x})q_4 + N_5^2(\mathbf{x})q_5, \end{aligned}$$

where N_i^e is the shape function of node i in element Ω^e . Interpolation with barycentric coordinates fulfills this condition because a quantity is linearly interpolated between the three vertices of a face.

Let $\mathbf{N}^e = (N_1^e, N_2^e, N_3^e, N_4^e)^T$ and $\mathbf{q}^e = (q_1, q_2, q_3, q_4)^T$, then the interpolation of q can be written concisely as

$$\bar{q}(\mathbf{x}) = \mathbf{N}^e(\mathbf{x})^T \mathbf{q}^e.$$

If we are interested in interpolating a vector quantity like the displacements \mathbf{u} , it is easiest to collect all twelve degrees of freedom in a vector

$$\mathbf{u}^e = (u_{1x}, u_{1y}, u_{1z}, u_{2x}, u_{2y}, u_{2z}, u_{3x}, u_{3y}, u_{3z}, u_{4x}, u_{4y}, u_{4z})^T$$

Ω^e , like \mathbf{u} and $\dot{\mathbf{u}}$, with the interpolated versions. We can do this on a per-element basis, which yields the per-element strain energy

$$\bar{\mathcal{U}}^e = \int_{\Omega^e} \frac{1}{2} \bar{\boldsymbol{\varepsilon}}^T \mathbf{E} \bar{\boldsymbol{\varepsilon}} dV = \frac{1}{2} (\mathbf{u}^e)^T \left(\int_{\Omega^e} \mathbf{B}^T \mathbf{E} \mathbf{B} dV \right) \mathbf{u}^e = \frac{1}{2} (\mathbf{u}^e)^T \mathbf{K}^e \mathbf{u}^e, \quad (2.21)$$

where \mathbf{K}^e is referred to as the element stiffness matrix.

For the per-element kinetic energy, we can define an approximate version in a similar fashion:

$$\bar{\mathcal{T}}^e = \int_{\Omega^e} \frac{1}{2} \rho \dot{\mathbf{u}}^T \dot{\mathbf{u}} dV = \frac{1}{2} (\dot{\mathbf{u}}^e)^T \left(\int_{\Omega^e} \rho \mathbf{N}^T \mathbf{N} dV \right) \dot{\mathbf{u}}^e = \frac{1}{2} (\dot{\mathbf{u}}^e)^T \mathbf{M}^e \dot{\mathbf{u}}^e, \quad (2.22)$$

where \mathbf{M}^e is known as the element mass matrix. The definitions of

$$\mathbf{K}^e = \int_{\Omega^e} \mathbf{B}^T \mathbf{E} \mathbf{B} dV, \quad \mathbf{M}^e = \int_{\Omega^e} \rho \mathbf{N}^T \mathbf{N} dV$$

are the vital accomplishments so far, and most of the effort in evaluating a finite element model goes into computing the element stiffness matrices and element mass matrices in an assemblage.

Now we shift our attention from a single element to the assemblage of elements that constitute the whole domain Ω . The element stiffness matrices \mathbf{K}^e pertaining to the individual elements can be assembled into a master stiffness matrix \mathbf{K} . This process corresponds to calculating the sums in Eqs. (2.19) and (2.20). The element stiffness matrices cannot be added up directly because each \mathbf{K}^e corresponds to a different vector \mathbf{u}^e that lists the degrees of freedom of this particular element. To solve this problem we define the vector $\tilde{\mathbf{u}}$, which lists the degrees of freedom of all nodes in the assemblage

$$\tilde{\mathbf{u}} = (u_{1x}, u_{1y}, u_{1z}, u_{2x}, u_{2y}, u_{2z}, \dots, u_{nx}, u_{ny}, u_{nz})^T,$$

where n is the total number of nodes.

Then we define expanded $3n$ -by- $3n$ element stiffness matrices $\tilde{\mathbf{K}}^e$ by pasting the non-zero elements of the 12-by-12 matrix \mathbf{K}^e into the locations of $\tilde{\mathbf{K}}^e$ that correspond to the degrees of freedom of this element. For the expanded element stiffness matrix it holds that

$$(\mathbf{u}^e)^T \mathbf{K}^e \mathbf{u}^e = \tilde{\mathbf{u}}^T \tilde{\mathbf{K}}^e \tilde{\mathbf{u}}.$$

This expansion process is illustrated in Fig. 2.9 using the example of a triangular element in a two-element assemblage.

The master stiffness matrix and—analogously—the master mass matrix are then defined by

$$\mathbf{K} = \sum_{e=1}^s \tilde{\mathbf{K}}^e, \quad \mathbf{M} = \sum_{e=1}^s \tilde{\mathbf{M}}^e.$$

Finally, the finite element approximation of the principle of stationary energy is

$$\delta(\bar{\mathcal{U}} + \bar{\mathcal{T}}) = \delta \left(\frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{u} + \frac{1}{2} \dot{\mathbf{u}}^T \mathbf{M} \dot{\mathbf{u}} \right) = 0.$$

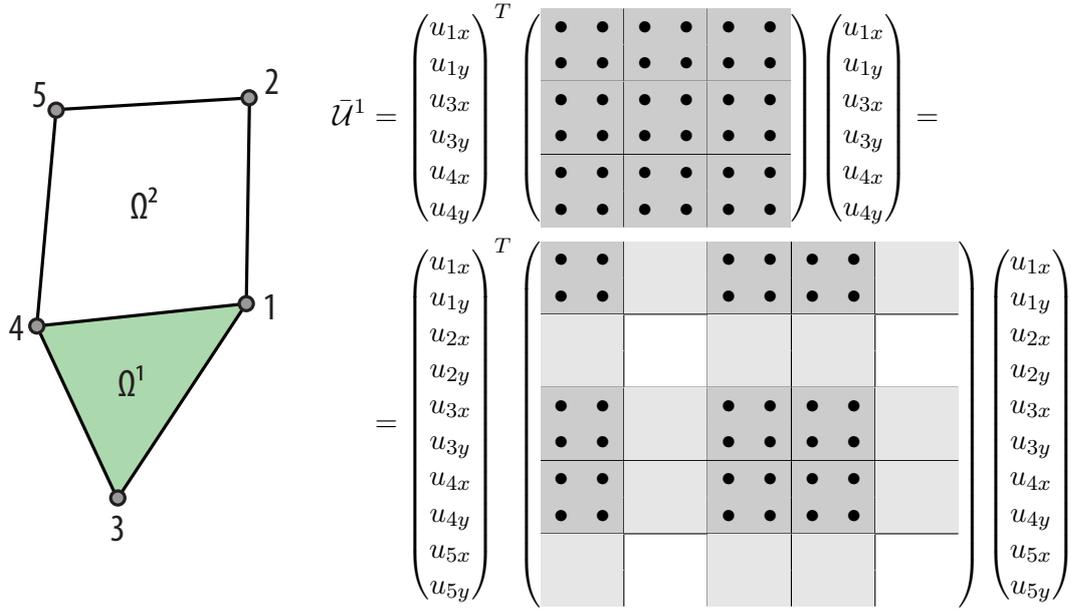


Figure 2.9: A 2d triangular element with corner nodes 1, 3, and 4 in a 5-node assemblage. The 6-by-6 element stiffness matrix is expanded to a 10-by-10 matrix by stamping the non-zero entries into the appropriate locations. The rows and columns in the expanded matrix that correspond to nodes 1, 3, and 4 are highlighted in gray.

We can make the transformations

$$\delta \bar{\mathcal{U}} = \frac{\partial \bar{\mathcal{U}}}{\partial \mathbf{u}} \delta \mathbf{u} = \frac{\partial \left(\frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{u} \right)}{\partial \mathbf{u}} \delta \mathbf{u} = (\mathbf{K} \mathbf{u})^T \delta \mathbf{u}$$

and, using an argument similar to Eq. (2.15),

$$\delta \bar{\mathcal{T}} = \frac{\partial \bar{\mathcal{T}}}{\partial \dot{\mathbf{u}}} \delta \dot{\mathbf{u}} = \frac{\partial \left(\frac{1}{2} \dot{\mathbf{u}}^T \mathbf{M} \dot{\mathbf{u}} \right)}{\partial \dot{\mathbf{u}}} \delta \dot{\mathbf{u}} = (\mathbf{M} \dot{\mathbf{u}})^T \delta \dot{\mathbf{u}} = (\mathbf{M} \ddot{\mathbf{u}})^T \delta \mathbf{u}.$$

A detailed derivation of the last equality can be found in the pertinent literature [MEH12]. Inserting back into Eq. (2.3) yields

$$\delta(\bar{\mathcal{U}} + \bar{\mathcal{T}}) = (\delta \mathbf{u})^T (\mathbf{K} \mathbf{u} + \mathbf{M} \ddot{\mathbf{u}}) = 0.$$

Because the virtual displacements $\delta \mathbf{u}$ are arbitrary, it must hold that

$$\mathbf{K} \mathbf{u} + \mathbf{M} \ddot{\mathbf{u}} = 0. \quad (2.23)$$

These are the finite element equations of motion for a free deformable solid that is not subject to any external forces. Free in this context means that the body is not supported in any point, i.e., no degrees of freedom are restricted.

2.4 Solving the Equations of Motion

The equations of motion (2.23) form a linear system of second-order ordinary differential equations. A system like this is known to have particular solutions of the form

$$\mathbf{u}^p(t) = \mathbf{v}e^{\lambda t}$$

and a general solution of the form

$$\mathbf{u}(t) = c_1 \mathbf{u}_1^p(t) + \dots + c_n \mathbf{u}_n^p(t),$$

where c_i are arbitrary constants and \mathbf{u}_i^p are linearly independent particular solutions.

Starting with $\mathbf{u}(t) = \mathbf{v}e^{\lambda t}$ as a trial solution, we find that

$$\ddot{\mathbf{u}}(t) = \lambda^2 \mathbf{v}e^{\lambda t}.$$

Substitution into the equations of motion yields

$$-\lambda^2 \mathbf{M} \mathbf{v} e^{\lambda t} = \mathbf{K} \mathbf{v} e^{\lambda t}$$

and, after division by the non-zero term $e^{\lambda t}$,

$$-\lambda^2 \mathbf{M} \mathbf{v} = \mathbf{K} \mathbf{v}. \quad (2.24)$$

This is a generalized eigenvalue problem with eigenvalues $-\lambda^2$ and eigenvectors \mathbf{v} .

From the definitions in Section 2.3, the following properties of \mathbf{M} and \mathbf{K} can be inferred:

- The per-element kinetic energy $\bar{\mathcal{T}}^e$, see Eq. (2.22), is quadratic in $\dot{\bar{\mathbf{u}}}$ and therefore non-negative. It follows that \mathbf{M}^e and, by summation, \mathbf{M} are positive semi-definitive.
- The same argument also ensures that \mathbf{M} is symmetric.
- The per-element strain energy $\bar{\mathcal{U}}^e$, see Eq. (2.21), is defined via the quadratic form $\bar{\boldsymbol{\varepsilon}}^T \mathbf{E} \bar{\boldsymbol{\varepsilon}}$ with the symmetric matrix \mathbf{E} . Therefore \mathbf{K}^e and \mathbf{K} are also symmetric.
- To show that \mathbf{K} is positive semi-definite as well, it is sufficient to show that the stiffness tensor \mathbf{E} is positive semi-definitive. This can be shown for isotropic materials, see Eq. (2.9), by invoking that Poisson's ratio ν is between -1 and 0.5 and that Young's Modulus E is positive.

These properties ensure that all eigenvalues $-\lambda^2$ in Eq. (2.24) are real and non-negative. It follows that $\lambda = i\omega$ where i is the imaginary unit and $\omega \in \mathbb{R}_0^+$.

A particular solution can be written as

$$\mathbf{u}(t) = \mathbf{v}e^{i\omega t} = \mathbf{v}(\cos \omega t + i \sin \omega t).$$

Both the real and the imaginary part of $\mathbf{u}(t)$ are real solutions to the original generalized eigenvalue problem:

$$\mathbf{u}_1(t) = \mathbf{v} \cos \omega t, \quad \mathbf{u}_2(t) = \mathbf{v} \sin \omega t.$$

We recognize these solutions as undamped oscillations with frequencies of $\frac{\omega}{2\pi}$.

It is interesting to note that the smallest eigenvalue is always zero and has a multiplicity of six. The corresponding solutions describe the six rigid-body modes, three translations and three rotations, of the deformable body. These modes are present because no DoF were restricted by introducing fixed supports into the system. Starting from the seventh eigenvalue, the solutions describe actual vibrations with increasing frequencies. The eigenvector that corresponds to a vibrational mode lists the geometric nodal displacements for the particular vibration, referred to as the mode shape. Examples for mode shapes are illustrated in Fig. 5.6. The application of finite element analysis in order to find oscillations of a system is usually referred to as finite element modal analysis.

2.5 Numerical Integration

In Section 2.3, we derived integral expressions for the element stiffness matrix and the element mass matrix such as

$$\mathbf{K}^e = \int_{\Omega^e} \mathbf{B}^T \mathbf{E} \mathbf{B} dV, \quad \mathbf{M}^e = \int_{\Omega^e} \rho \mathbf{N}^T \mathbf{N} dV.$$

For the toy example developed in Section 2.3, the integrand in the definition of \mathbf{K}^e is constant, and the integrand in the definition of \mathbf{M}^e is a polynomial in the variable of integration \mathbf{x} . Section 4.3 will present the derivation of more complex elements, and it will become clear that the integrands are not necessarily polynomials in \mathbf{x} but can be rational functions, i.e., fractions of polynomials in \mathbf{x} . While all rational functions can be integrated analytically in theory, it is computationally expensive to do so. Instead, finite element codes usually use numerical integration techniques to evaluate element matrices.

There are two classes of numerical integration procedures that can be used for the evaluation of \mathbf{K}^e and \mathbf{M}^e . Both classes are based on the idea of approximating an integral over a function with a finite weighted sum of function values

$$\int_0^1 f(x) dx \approx \sum_{i=1}^n w_i f(x_i). \quad (2.25)$$

Any quadrature rule can be expressed by specifying the number of samples n , the locations of the sample points x_i , and the weighting scheme w_i . The simpler and less powerful method of the two uses the Newton–Cotes quadrature rules. There is one Newton–Cotes rule for each $n > 1$. The distribution of the sample points x_i is equidistant in the interval $[0, 1]$, i.e.,

$$x_1 = 0, x_2 = \frac{1}{n-1}, x_3 = \frac{2}{n-1}, \dots, x_{n-1} = \frac{n-2}{n-1}, x_n = 1.$$

The weights w_i are chosen such that the n -point Newton–Cotes rule integrates polynomials with a degree of up to $(n-1)$ exactly. This condition can be formulated as a set of linear

equations in w_1, \dots, w_n

$$\begin{aligned}
 f(x) = 1 &: \sum_{i=1}^n w_i = \int_0^1 1 \, dx = 1, \\
 f(x) = x &: \sum_{i=1}^n w_i \frac{i-1}{n-1} = \int_0^1 x \, dx = \frac{1}{2}, \\
 f(x) = x^2 &: \sum_{i=1}^n w_i \frac{(i-1)^2}{(n-1)^2} = \int_0^1 x^2 \, dx = \frac{1}{3}, \\
 &\vdots \\
 f(x) = x^{n-1} &: \sum_{i=1}^n w_i \frac{(i-1)^{n-1}}{(n-1)^{n-1}} = \int_0^1 x^{n-1} \, dx = \frac{1}{n},
 \end{aligned} \tag{2.26}$$

which uniquely determine the weights. Note that this derivation procedure works even if the sample locations are not spaced regularly. The only requirement is that the n sample points are distinct and lie in the interval $[0, 1]$.

As an example, the three-point Newton–Cotes rule is derived here. It has the sample point locations $x_1 = 0$, $x_2 = \frac{1}{2}$, $x_3 = 1$, and the weights can be found by solving the linear equation system

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & \frac{1}{2} & 1 \\ 0 & \frac{1}{4} & 1 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \end{pmatrix},$$

which yields $w_1 = \frac{1}{6}$, $w_2 = \frac{2}{3}$, $w_3 = \frac{1}{6}$. This rule integrates constant, linear, and quadratic functions exactly and is also known as Simpson’s rule. Fig. 2.10a shows an example in which the rule is applied to a fifth-order polynomial.

The second and more powerful numerical integration method follows the Gauss quadrature rules. These rules also have the basic form Eq. (2.25), but they achieve higher accuracy than the Newton–Cotes rules. The n -point Gauss quadrature rule integrates polynomials with a degree of up to $(2n-1)$ exactly as opposed to a degree of up to $(n-1)$. This is made possible by replacing the regular sampling method of the Newton–Cotes formulae with a set of optimized, non-equidistant sampling locations.

Gauss quadrature rules are derived using the sequence of Legendre polynomials $P_n(x)$. The n -th Legendre polynomial has degree n and is defined by Rodrigues’ formula

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n.$$

It has two special properties that are needed in the derivation of Gauss quadrature rules. Firstly, it holds that

$$\int_{-1}^1 x^k P_n(x) \, dx = 0, \quad k \in \{0, 1, \dots, n-1\}.$$

Secondly, all n roots of the n -th Legendre polynomial are distinct, real, and lie in the interval $[-1, 1]$. From the first property, it follows that $P_n(x)$ is orthogonal to any

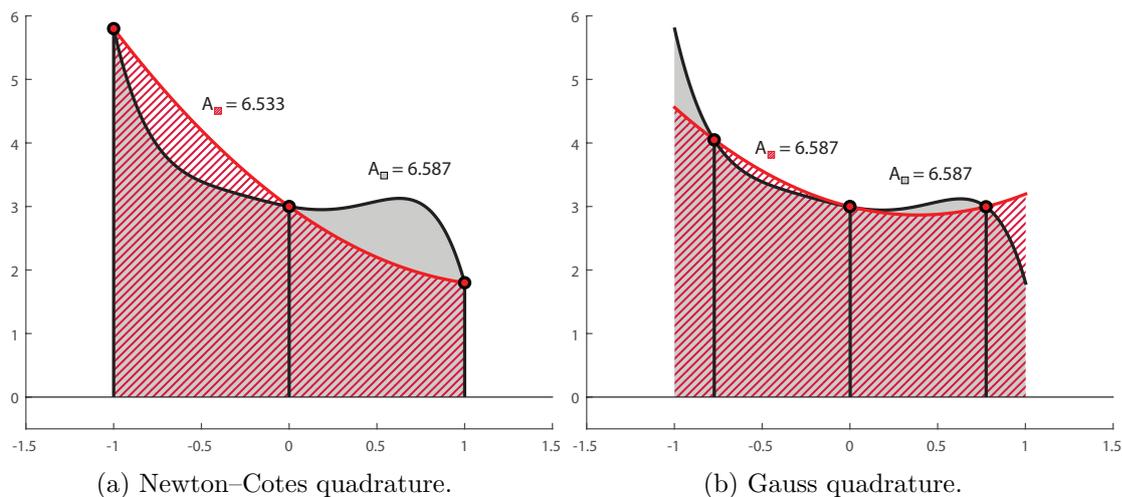


Figure 2.10: Three-point quadrature of a fifth-order polynomial with (a) the Newton–Cotes formula and (b) the Gauss formula. The area of the hatched region results from the respective quadrature rule. The Newton–Cotes formula approximates the integral, while Gauss quadrature finds the exact result.

polynomial $q(x)$ of degree $(n - 1)$ or lower. To prove this, let $q(x) = \sum_{k=0}^{n-1} q_k x^k$. Then it holds that

$$\int_{-1}^1 q(x) P_n(x) dx = \sum_{k=0}^{n-1} q_k \int_{-1}^1 x^k P_n(x) dx = 0. \quad (2.27)$$

In order to derive the sample positions and weights for n -point Gauss quadrature, let $f(x)$ be a polynomial with a degree of at most $(2n - 1)$. Dividing $f(x)$ by $P_n(x)$ yields a quotient $q(x)$ and a remainder $r(x)$, both of which are polynomials with a degree of at most $(n - 1)$. We can write this division as

$$f(x) = q(x)P_n(x) + r(x).$$

Using Eq. (2.27), it follows that

$$\int_{-1}^1 f(x) dx = \int_{-1}^1 q(x)P_n(x) dx + \int_{-1}^1 r(x) dx = \int_{-1}^1 r(x) dx.$$

Since $r(x)$ is of degree $(n - 1)$ or lower, it can be integrated exactly using an n -point quadrature rule for any choice of n distinct sampling locations. The corresponding sample weights can always be found using the linear equation system in Eq. (2.26).

For Gauss quadrature, the sample point locations x_1, \dots, x_n are set to the roots of the Legendre polynomial $P_n(x)$, which are guaranteed to be real, distinct, and lie in the interval $[-1, 1]$. Let w_1, \dots, w_n denote the weights corresponding to these sample point

locations. This allows us to write

$$\begin{aligned}\sum_{k=1}^n w_k f(x_k) &= \sum_{k=1}^n w_k q(x_k) \underbrace{P_n(x_k)}_0 + \sum_{k=1}^n w_k r(x_k) = \sum_{k=1}^n w_k r(x_k) = \\ &= \int_{-1}^1 r(x) dx = \int_{-1}^1 f(x) dx.\end{aligned}$$

As an example, we derive the 3-point Gauss quadrature rule, which integrates polynomials of degree 5 or lower exactly. The third Legendre polynomial is defined as

$$P_3(x) = \frac{1}{2}(5x^3 - 3x)$$

and has roots

$$x_1 = -\sqrt{\frac{3}{5}}, \quad x_2 = 0, \quad x_3 = \sqrt{\frac{3}{5}},$$

which serve as sample point locations. To find the corresponding weights, we impose the condition that polynomials of degree 2 or less are integrated exactly. This yields the equations

$$\begin{aligned}f(x) = 1 : \quad w_1 + w_2 + w_3 &= \int_{-1}^1 1 dx = 2, \\ f(x) = x : \quad -\sqrt{\frac{3}{5}}w_1 + \sqrt{\frac{3}{5}}w_3 &= \int_{-1}^1 x dx = 0, \\ f(x) = x^2 : \quad \frac{3}{5}w_1 + \frac{3}{5}w_3 &= \int_{-1}^1 x^2 dx = \frac{2}{3}.\end{aligned}$$

Solving the linear equation system

$$\begin{pmatrix} 1 & 1 & 1 \\ -\sqrt{\frac{3}{5}} & 0 & \sqrt{\frac{3}{5}} \\ \frac{3}{5} & 0 & \frac{3}{5} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ \frac{2}{3} \end{pmatrix}$$

yields the weights $w_1 = \frac{5}{9}$, $w_2 = \frac{8}{9}$, $w_3 = \frac{5}{9}$. Fig. 2.10b shows an example in which the rule is applied to a fifth-order polynomial.

The Newton–Cotes rules were introduced to integrate functions over the interval $[0, 1]$, and the Gauss quadrature rules to integrate functions over the interval $[-1, 1]$. Both rule families can easily be adapted to arbitrary intervals $[a, b]$ by performing a coordinate transformation. An integral over $[a, b]$ can be transformed into an integral over $[-1, 1]$ using the transformation

$$y = \frac{b-a}{2}x + \frac{a+b}{2}.$$

This yields the Gauss quadrature rule

$$\int_a^b f(x) dx \approx \frac{b-a}{2} \sum_{k=1}^n w_k f\left(\frac{b-a}{2}x_k + \frac{a+b}{2}\right).$$

In finite element models for applications in solid mechanics, we typically encounter surface integrals and volume integrals. The quadrature rules introduced so far integrate one-dimensional functions, but they can be generalized to higher dimensions. In order to evaluate a surface integral over a quadrilateral region or a volume integral over a hexahedral region, one can form tensor products of the original rule like so:

$$\int_{-1}^1 \int_{-1}^1 f(x, y) dx dy \approx \sum_{k=1}^n \sum_{l=1}^n w_k w_l f(x_k, x_l).$$

For quadrilateral regions, this yields quadrature rules with four points, nine points, etc.

The tensor product approach can also be used to integrate triangular regions by introducing a coordinate transformation that transforms the rectangular region into a triangle. However, by shrinking an edge of the rectangle to a point, the inverse Jacobian of the transformation will have a singularity there, and this decreases accuracy. There are several additional disadvantages of tensor product rules for triangles. They have an unnecessarily high number of integration points, and the sample point locations are not symmetric and usually concentrated around the vertex that has been shrunk from the edge of the rectangle.

The alternative is to derive triangle Gauss quadrature rules specially because they provide higher accuracy. Usually these rules are derived for integrals over the standard triangular element, which is bounded by the vertices $(0, 0)$, $(1, 0)$, $(0, 1)$. To derive a rule that integrates polynomials of degree 2 or lower exactly, i.e., the polynomials 1 , x , y , x^2 , xy , y^2 , we impose the six conditions

$$\begin{aligned} f(x, y) = 1 &: \sum_{k=1}^n w_k = \iint_{\Delta} 1 dA = \frac{1}{2}, \\ f(x, y) = x &: \sum_{k=1}^n w_k x_k = \iint_{\Delta} x dA = \frac{1}{6}, \\ &\vdots \\ f(x, y) = y^2 &: \sum_{k=1}^n w_k y_k^2 = \iint_{\Delta} y^2 dA = \frac{1}{12}. \end{aligned}$$

Choosing $n = 1$ yields three variables w_1 , x_1 , y_1 , which is not enough to solve the six equations. Choosing $n = 2$ yields six variables w_1 , w_2 , x_1 , x_2 , y_1 , y_2 , which has a solution, however not a symmetric one. Therefore, a 2-point quadrature rule for triangles is not commonly used. Choosing $n = 3$, the solution is no longer unique, but the most commonly used 3-point rule is

$$\begin{aligned} (x_1, y_1) &= \left(\frac{1}{6}, \frac{1}{6}\right), & (x_2, y_2) &= \left(\frac{2}{3}, \frac{1}{6}\right), & (x_3, y_3) &= \left(\frac{1}{6}, \frac{2}{3}\right) \\ w_1 &= \frac{1}{6}, & w_2 &= \frac{1}{6}, & w_3 &= \frac{1}{6}. \end{aligned}$$

In order to integrate complex three-dimensional domains like triangular prisms, it is possible to create tensor products between a triangle rule and a one-dimensional rule. This is useful to compute the element matrices of the wedge element introduced in Section 4.3.

In the integration of \mathbf{K}^e and \mathbf{M}^e , we use Gauss quadrature because of its accuracy advantages over Newton–Cotes formulae. Element domains are usually distorted geometric primitives, and, in order to integrate over them, they are transformed to a standard element domain with a change of variables. For example, quadrilateral elements are transformed into the standard square element $[-1, 1] \times [-1, 1]$. This involves multiplying the integrand with determinant of the Jacobi matrix of the transformation and turns the integrand into a rational function. The consequence is that the integral cannot be evaluated exactly using numerical integration. If the Jacobian is well behaved, i.e., if the element is not heavily distorted, Gauss quadrature still produces accurate results.

The major concern with the choice of an appropriate quadrature rule is that of rank sufficiency [Bat06]. The element mass matrix \mathbf{M}^e has full rank, and the element stiffness matrix \mathbf{K}^e has a rank deficiency that is equal to the number of rigid-body modes of the element. In the case of three-dimensional continuous elements for solid mechanics, the number of rigid-body modes is six—three translational and three rotational degrees of freedom. If the integral in the definition of an element matrix is evaluated with a quadrature rule, it is of paramount importance that the rank of the matrix be preserved.

To calculate the rank of an element matrix that is evaluated with n -point Gauss quadrature, we need to know the rank of the integrand. For the stiffness matrix \mathbf{K}^e of an element with m degrees of freedom, the integrand has the form $\mathbf{B}^T \mathbf{E} \mathbf{B}$, where B is a 6-by- m matrix and E is a full-rank 6-by-6 matrix. Therefore, the integrand $\mathbf{B}^T \mathbf{E} \mathbf{B}$ is an m -by- m matrix with a rank of six. Using n -point Gauss quadrature, the integral is replaced with a sum of n m -by- m matrices with a rank of six. The result of this summation is an m -by- m matrix with a rank of at most $6n$, and in practice this rank is attained exactly. It can be concluded that, in order to create a rank-sufficient element stiffness matrix, n has to be chosen such that $6n \geq m - 6$. For example, for a 20-node hexahedral element, the number of degrees of freedom $m = 60$, which gives an element stiffness matrix with a rank of 54. In order to ensure rank sufficiency, at least 9 quadrature sample points are required. The tensor product method yields a $2 \times 2 \times 2$ -point rule and a $3 \times 3 \times 3$ -point rule for hexahedral regions. Since the 8-point rule would compute an element stiffness matrix with a rank of 48, the 27-point rule has to be used.

The integrand in the definition of the element mass matrix \mathbf{M}^e has the form $\mathbf{N}^T \mathbf{N}$ where \mathbf{N} is a 3-by- m matrix. Therefore, the integrand has a rank of three. The element mass matrix has full rank, i.e., a rank of m . Since every summand in the Gauss quadrature rule adds three to rank of the result, n has to be chosen such that $3n \geq m$. For the example of the 20-node hexahedral element, the $3 \times 3 \times 3$ -rule fulfills this requirement since $3 \cdot 27 \geq 60$ and is the appropriate choice.

We illustrate the deleterious effect of choosing a Gauss quadrature rule with an insufficient number of sample points with a two-dimensional plane stress problem. Consider a

nine-node quadrilateral 2d element with 18 degrees of freedom. In two dimensions there are three rigid-body modes, two translational and one rotational, yielding an element stiffness matrix with a rank of 15. The appropriate Gauss quadrature rule is the tensor product of the three-point rule, which results in a nine-point rule. In the first scenario, \mathbf{K}^e is integrated with this rule, and \mathbf{M}^e is integrated with a sufficiently accurate rule as well. Plugging these matrices into the equations of motion yields

$$\mathbf{K}^e \mathbf{u} + \mathbf{M}^e \ddot{\mathbf{u}} = 0,$$

which reduces to a generalized eigenvalue problem as discussed in Section 2.4. Since \mathbf{K}^e has a rank deficiency of three, and \mathbf{M}^e has full rank, zero appears as an eigenvalue with a multiplicity of three. The corresponding eigenvectors describe linear combinations of the three rigid-body modes of the element. The remaining eigenvectors describe vibration modes, and the corresponding eigenvalues give their frequencies. The first eight natural modes are illustrated in Fig. 2.11.

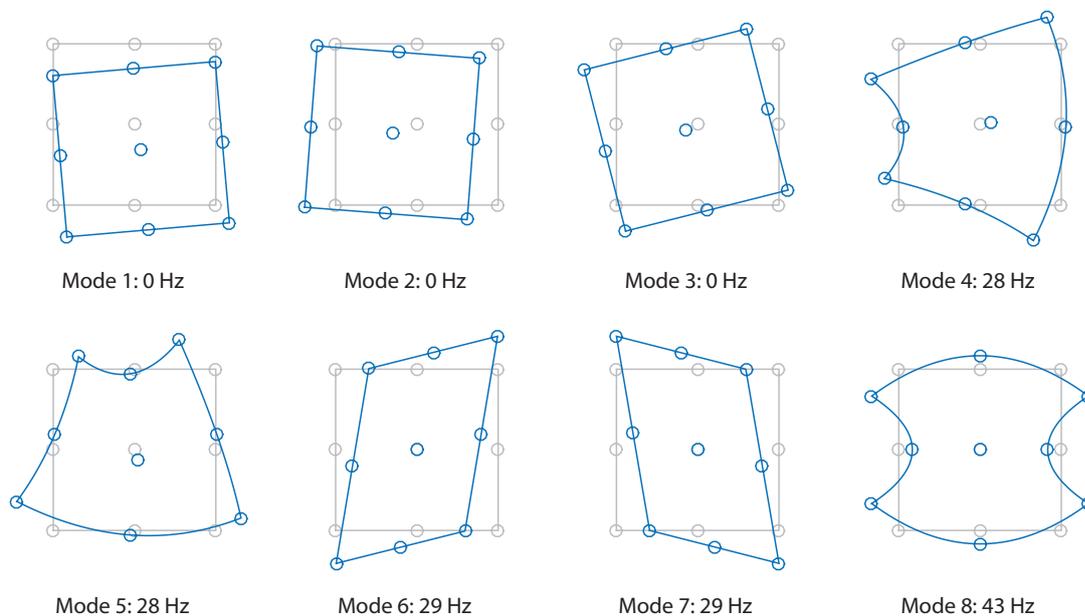


Figure 2.11: Eigenmodes of a nine-node quadrilateral element. Numerical integration has been performed with sufficient accuracy, which produces the correct number of three rigid-body modes. The undeformed state of the element is plotted in gray, the mode shapes in blue.

In the second scenario, we replace the nine-point quadrature rule for the stiffness matrix with a four-point rule. Since every summand adds three to the rank of the result, this yields a stiffness matrix with a rank of twelve. The consequence is that, when solving the equations of motion, zero appears as an eigenvalue with a multiplicity of six. Fig. 2.12 illustrates this phenomenon, which is known as spurious modes. A spurious mode is

a deformation shape that is falsely predicted to have zero strain energy by the finite element model. The introduction of spurious modes leads to non-physical results and has to be avoided.

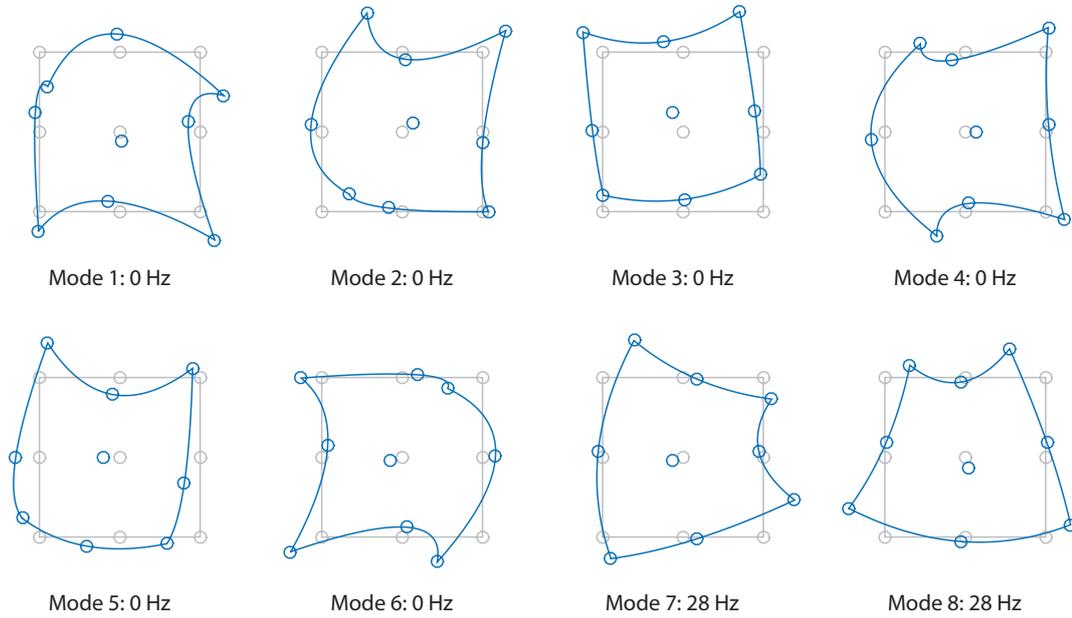


Figure 2.12: Spurious mode of a nine-node quadrilateral element. Numerical integration has been performed with insufficient accuracy, which produces too large a number of zero-strain-energy modes. The undeformed state of the element is plotted in gray, the mode shapes in blue.

Related Works

3.1 Shape Optimization

This section surveys literature about the optimization of shapes with respect to physical properties, appearance, producibility, or other desirable qualities. We will start with an overview of traditional shape optimization tasks in computer graphics and subsequently explore more recent works, especially those fueled by the wide availability of 3d printers.

A groundbreaking work in this area is Taubin's seminal publication on a signal processing approach to surface processing [Tau95]. The most fundamental contribution in this work is the definition of a Laplace operator on triangle meshes. This adapts the Laplace-Beltrami operator known from differential geometry to a discrete setting and creates natural analogues for concepts such as curvature and principal directions. The first application of this operator was surface smoothing or low-pass filtering. The umbrella operator proposed by Taubin is extended to take mesh geometry into account in subsequent publications [DMSB99], and the surface fairing algorithm is modified to support mesh boundaries and constraints. Meyer et al. [MDSB03] extend further concepts from differential geometry to triangle meshes. Nealen et al. [NISA06] apply these ideas for triangle shape optimization and feature preserving smoothing.

A different challenge in shape processing is the editing of shapes in an intuitive, predictable, and user-friendly way. The influential work by Alexa et al. [ACOL00] tackles shape interpolation—the gradual transformation from one shape into another in a physically plausible way. This task is posed as an energy minimization problem, where a high energy ensues if triangles are severely distorted during shape interpolation. Related works explore similar tasks such as manual surface deformation with handles and the transfer of surface details between different surfaces [SCOL⁺04, IMH05].

Many applications require a parametrization of a surface, i.e., a continuous mapping from a surface with boundary into the plane. The challenge is to find a mapping that distorts the surface as little as possible and does not produce any overlaps. The most successful approach is to define a distortion measure in the vertices of the flattened

surface and formulate an optimization problem in order to minimize it. Particularly efficient are algorithms that use linear constraints [DMA02, LPRM02, ZLS07], while non-linear formulations provide the best quality [SLMB05].

The spectral decomposition of the mesh Laplacian has been explored as a means of performing frequency band filtering and mesh compression [Lév06, VL08]. Most recently, Musialski et al. use manifold harmonics in order to represent the geometry of a surface with a small number of parameters [MAB⁺15], an approach well suited to shape optimization problems.

In the past five years, works about shape optimization have been focusing more and more on fabrication tasks, i.e., to create models with the intent of producing an actual physical copy. This trend has been brought about by the rise of cheap 3d printers that are now available to a broad audience. An early work in this area explores the non-linear material behavior of heterogeneous, soft bodies [BBO⁺10]. The authors develop an optimization algorithm that approximates the material behavior of a soft object by a stack of layers with varying microstructures. The search for an optimal set of layers is posed as a combinatorial optimization problem that explores a finite set of base material from which to choose the individual layers. In order to validate their results, a multi-material 3d printer that is capable of printing soft and hard materials is used to produce replicas. A recent work by Chen et al. [CZXZ14] tackles a different problem relating to the design and fabrication of elastic bodies. They assume a homogeneous, elastic material that is soft enough for any fabricated object to sag due to gravity and whose resting pose is appreciably different from its initial pose. The contribution of this work is a tool that allows for inverse shape design: The user specifies the desired resting pose as a 3d model, and a corresponding initial pose is calculated automatically.

In the same vein, Skouras et al. [STBG12] optimize the initial shape of a rubber balloon such that it assumes the desired shape in its inflated state. A later work explores the design of balloons made from planar, inextensible membranes [STK⁺14]. Their algorithm finds a partition of the desired shape into near-flat pieces and simulates the inflated state using an advanced formulation of strain energy.

A different emergent research direction is the optimization of kinetic systems that follow a prescribed motion. Ceylan et al. [CLM⁺13] represent a character as a hierarchical structure of rigid links, i.e., a skeleton. A periodic motion sequence is prescribed as time-varying rotations between connected links and translated into an assemblage of gears and pulleys. Their system can be used to optimize in-plane motions such as walking and running. Coros et al. [CTN⁺13] prescribe motions as curves specified at the endpoints of limbs. These curves are matched by gear mechanisms by browsing a database of possible motions that has been created beforehand. All resulting parts, e.g., limbs, gears, links, and support structures, are produced individually with a 3d printer, thereby creating a large design space. Thomaszewski et al. [TCG⁺14] enable the design of characters that consist only of rigid links and joints that are driven by a single input crank. Their design tool provides exploration options for different linkage topologies while automatically optimizing the current design to match a given motion.

A big advantage of additive manufacturing techniques, such as fused-deposition

modeling (FDM) printing, is that intricate internal structures can be printed at no additional cost. Cali et al. [CCA⁺12] utilize this property to fabricate fully poseable character models. Their joints are designed to allow the desired angular freedom, and the whole model can be printed in one piece. Bächer et al. [BBJP12] automatically estimate suitable joint locations and geometry from a skinned mesh. Skouras et al. [STC⁺13] make articulation of their characters possible by distributing hard and soft materials throughout the interior such that limbs remain rigid but the pose can be adjusted.

Natural concerns that arise during the fabrication of physical artifacts are those regarding stability, producibility, and production costs. Stava et al. [SVB⁺12] address the stability question by performing finite element stress analysis. If structurally weak regions are detected, the algorithm tries to reinforce the mesh by thickening fragile parts, hollowing the interior of the object to reduce its dead load, or adding support struts. The work by Wang et al. [WWY⁺13] is concerned with reducing the printing costs of an object while retaining structural soundness and printability. Their algorithm hollows out the object and adds a frame structure under the skin for reinforcement. A constraint system ensures quality criteria like stability, shape, and balance. The same problem is also tackled by Lu et al. [LSZ⁺14], who propose to introduce a number of air bubbles into the object. The resulting structures are reminiscent of a Voronoi diagram in three dimensions and possess a high strength-to-weight ratio. Zhou et al. [ZPZ13] take a purely analytic viewpoint and develop a framework to assess the stability of an object without any specific assumptions regarding the loads. Weak regions are extracted by performing modal analysis and searching for high-stress regions in the presence of vibrations. The algorithm can then determine a set of worst-case loads that will cause the highest stress possible for each weak region. The designer can subsequently decide whether these worst-case scenarios are likely and whether the design needs an overhaul. Dumas et al. [DHL14] address a difficulty specific to the FDM printing process. During fused deposition modeling, layers of melted plastic are deposited on top of already existing layers, and therefore it is impossible to print stalactites or even strong overhangs. Print software currently in use solves this issue by printing dense support structures wherever a model needs support, thereby significantly increasing material consumption. A more efficient solution is to use a sparse support structure akin to scaffolding used on construction sites—a combination between vertical columns and horizontal bridges.

The line of research most inspiring to this thesis follows the publication by Prévost et al. [PWLSH13] on the optimization of an object’s center of gravity. Their goal is to balance a given mesh by moving the center of gravity to a position that projects downwards onto the base of support, i.e., the convex hull of all points touching the ground. This is achieved by voxelizing the interior of the object and iteratively removing voxels as a plane sweeps across the object. If static equilibrium cannot be achieved with this step alone, the shape is automatically deformed using deformation handles that the user places manually. A considerable generalization of this idea is made by Bächer et al. [BWBSH14]. They address the optimization of general mass properties, such as mass, center of gravity, and the tensor of inertia. Their framework is also based on a voxelization of the interior, but the optimization routine searches for more general voxel

configurations. Using this method, it is possible to design yo-yos and spinning tops with strongly asymmetric shapes. Musialski et al. [MAB⁺15] propose another solution for the optimization of mass properties. Given an outer surface, they find an inner outer surface such that the enclosed solid satisfies the prescribed mass properties. Their formulation of the optimization problem is fully continuous, and the resulting surfaces are always smooth.

3.2 Sound Synthesis

The discipline of sound synthesis has the goal of supplementing a virtual scene with a soundtrack that is computer generated instead of recorded. Techniques to infer sound from a digital representation of a physical process is an alternative to recording a variety of sounds and attaching them to objects or events in a computer animation, a film, or a video game. This process has the potential to be more flexible if the medium allows for a high degree of interactivity—not every possible interaction between the medium and the user may be foreseen and an appropriate sound prepared. We give an overview of publications in this area because it bears a resemblance to the goal of thesis: Sound synthesis infers sounds from objects, while we find objects to match sounds.

Takala et al. [TH92] present a variety of methods to synchronize a soundtrack with an animation in a semi-automatic manner. They propose several sound sources, e.g., digitally recorded sounds for impacts, natural mode banks, which were previously used for animation [PW89], and procedural sounds. Attaching sounds to objects and propagating sounds inside a scene is performed in analogy to texture mapping and coordinate space transformation. Another early work in this area explores the role of sound in computer interfaces by drawing analogies between everyday events and actions performed in, e.g., a file explorer [Gav93]. The focus lies on parametrization of the sound models according to human perception, such that sounds give meaningful acoustic feedback about the action performed by the user.

An important contribution to physically based sound synthesis is the idea of a sound map [vdDP96, vdDP98]. The vibration patterns of an object are approximated by a wave function and precomputed analytically, which restricts the method to simple geometries. This yields a number of eigenfunctions, each of which defines an amplitude distribution over the object. The set of these amplitude distributions constitute the sound map, which describes the eigenmodes that will be excited if there is an impact at a particular location. This idea is extended upon in subsequent works [vdDKP01, PDJ⁺01] to include scraping sounds and rolling sounds. Scraping is procedurally modeled as fractal noise, where the fractal dimension is varied to account for the roughness of a surface.

O’Brien et al. [OCE01] show how to implement sound synthesis into a solid body simulator, e.g., a finite element simulation routine [OH99]. This method is very direct in the sense that it does not precompute eigenmodes but derives pressure waves directly from the motions of an object. A requirement is that the simulator perform time step integration on the order of 10^{-5} seconds such that vibrations in the audible spectrum are captured. Naturally, this method is restricted to offline computation, but it is extended to

work with real-time rigid-body simulation software in a subsequent publication [OSG02]. In this work, the finite element equations of motion are decomposed into a set of decoupled oscillators. The impact forces derived from the rigid-body simulation are then projected onto the vibrational modes in real time.

Van den Doel et al. [vdDKP04] propose an algorithm to reduce the computational load of sound synthesis in a scenario involving many collisions. This is achieved by applying psychoacoustic models to predict which impacts will be inaudible to the listener in order to mask them. Raghuvanshi et al. [RL06] model the elastic properties of a solid with a mass-spring model and introduce more optimizations to handle many impact sounds in complex environments, most notably the fusion of similar modes. Bonneel et al. [BDT⁺08] further accelerate sound synthesis by summing the contributions of all impact sounds in the frequency domain to avoid the evaluation of every acoustic response in favor of a single FFT computation.

James et al. [JBP06] increase the quality of synthesized sounds by taking into account indirect effects, such as the occlusion and reflection of a sound wave that originates at a specific location on an object. These effects of wave propagation are simulated for larger scenes by Raghuvanshi et al. [RSM⁺10]. Chadwick et al. [CAJ09] are the first to use non-linear coupling between vibrational modes to simulate impact sounds on thin-shelled solids such as cymbals or plastic containers. This creates characteristic pitch changes as the sound develops.

A useful technique for contact sound modeling such as scraping and rolling was developed by Ren et al. [RYL10]. They use roughness information from a normal map texture to simulate impacts that are not apparent from the geometry of colliding objects. The works by Zheng et al. [ZJ10, ZJ11] propose solutions to model sounds for complex fracture animations and contact phenomena such as rattling sounds. Chadwick et al. [CZJ12] synthesize contact sounds that result from very brief collisions and abrupt decelerations that occur if two brittle objects collide.

Langlois et al. [LAJJ14] propose a scheme to compress eigenmode information for complex scenes. Each eigenmode is represented by a number of sample points from which the eigenmode can be reconstructed via moving least squares. The storage requirements are further reduced by exploiting the self-symmetry of modes and neighboring modes that form rotationally congruent pairs. The inverse problem of contact sound synthesis is the inference of a physically plausible animation given a series of sounds [LJ14]. This problem is tackled by deriving a set of impact and sliding events from the audio track and trying to finding a plausible fit from a database of sampled rigid-body animations.

3.3 Instrument Analysis

The millennia-old craft of instrument creation has always been a laborious process of trial-and-error, and what sets the sound of a Stradivari apart from any other violin largely remains a mystery. Nevertheless, there are accurate physical models that describe the behavior of sound waves and solid bodies, as well as good approximations such as the finite element modal analysis framework described in Chapter 2. There are a large

number of works about instrument analysis by the engineering and acoustics communities, and this section specifically reviews publications that make use of finite element modal analysis. The physical counterpart of finite element modal analysis is dubbed experimental modal analysis. This process finds natural mode shapes and frequencies by attaching accelerometers to a physical object and measuring the vibrations that ensue after a hit with an impulse hammer. The accuracy of a finite element model is often validated using this technique.

A comparison between experimental modal analysis and finite element modal analysis is conducted on a marimba bar, a rectangular bar with an undercut made out of Rosewood [BCT⁺99]. The authors use three layers of brick elements in the finite element mesh that represents the marimba bar. Initially, an isotropic material model is used, but it only achieves accordance between the FEM solution and the experimental values for the lowest frequencies. Using an orthotropic material model to account for the primary direction of growth yields an agreement error of less than four percent for the first twelve modes. Bretos et al. [BSM99a] study the influence of differently shaped undercuts and the variation of different material parameters on the natural mode spectrum of marimba bars. The primary concern is the ratio between the first few natural frequencies, because harmonic ratios, e.g., 1:3 or 1:4, lead to better sound quality. The authors find that a parabolic undercut is most suitable to achieve this goal. The other finding is that the Young's Modulus parallel to the grain of wood, the shear moduli parallel to the grain, and the density of the material have the biggest influence on the natural modes.

Curtu et al. [CSCR09] analyze the mode shapes of classical guitar bodies. Guitar bodies differ greatly in the type of bracing, i.e., the system of wooden struts attached to the top plate from the inside, which is used to support the plate against the tension of the strings. It is concluded that the strut layout mostly influences the quality of high-frequency modes while lower frequencies remain unaffected.

The instrument most studied and most elusive in acoustic character is the violin. The violin box has a complex shape, and luthiers pay close attention to the carvings to achieve the highest sound quality and visual appeal. Bretos et al. [BSM99b] attempt finite element analysis of the assembled violin body and achieve great correspondence with experimental modal analysis. They also describe the steps of the violin tuning process and assess the eigenmode shapes after every step using FEM. Yu et al. [YJKK10] model a violin top plate using shell elements. They attempt to optimize the thickness distribution of the elements in order to achieve ideal nodal line shapes, i.e., the curves on the plate that remain at rest for a given mode. A recent work on violin top plates explores the effect of using composite materials, e.g., balsa and carbon fiber instead of traditional materials like spruce, on the natural frequency spectrum [Lu13].

Umetani et al. [UMI10] develop a tool that allows the user to design the bars for a metallophone, e.g., a xylophone. The user manipulates the boundary curve of a planar shape, and the application triangulates it and predicts the frequency of the pitch at interactive rates. The finished design is fabricated by cutting the bar shapes from a metal plate. If the actual frequencies do not coincide with the FEM predictions precisely, the pieces are manually adjusted by trimming the edges.

Optimization of Natural Frequencies

4.1 Overview

This chapter describes the computational framework that we use to synthesize the geometry of a solid that will exhibit the desired pitch if fabricated from the targeted material. The framework can be broken down into three modules: the definition of a design space, the evaluation of a given design using finite element modal analysis, and computational shape optimization to find a feasible design.

We choose our design space such that, on the one hand, it can reproduce common shapes in the family of instruments that we want to model and, on the other, it can be parametrized with a small set of optimization variables. To account for varying degrees of shape complexity and difficulty of optimization, three different design-space parametrizations are proposed.

For the evaluation of natural frequencies using the finite element method, we develop a number of 3d element types that can be used to describe the geometry of the design. We discuss a method to derive shape functions for continuous elements and choose methods of integration. The structural thin-shell elements are introduced briefly as an alternative to continuous element types. We conclude this section with remarks about Jacobian positiveness and a comparison of the computational efforts for different elements.

The last section discusses the formulation of the optimization problem and the constraints for different design-space parametrizations.

4.2 Design Space

The design space of an optimization problem is the family of all feasible solutions that can be produced by varying the optimization parameters. The goal of this work is the

fabrication of solids, and therefore our design space is a subset of the family of 2-manifolds, which describe the boundary of a solid. In particular, we can identify the following design goals:

- The main objective is the control of the lowest natural frequency (pitch) of the volume enclosed by the surface. Since the pitch depends on the shape and the material of an object, the material parameters and the desired pitch are constants that the user can specify beforehand.
- The approximate shape of the resulting object can be determined by the user.
- The design space should be general enough to allow for a wide range of frequencies, at the very least one octave.

The formulation of the optimization problem should also be subject to the following considerations:

- The number of optimization parameters should be as low as possible to keep the run-time of each optimization iteration short.
- The formulation of the optimization problem should be amenable to an off-the-shelf non-linear optimization routine.
- Additional design goals like the optimization of harmonic ratios between overtone frequencies can be enforced.

4.2.1 Motivation

We begin our analysis of the design goals by a survey of instruments that belong to the class of struck idiophones. Instruments of this category are characterized by the way they produce sounds: They are struck by an object and hence start vibrating as a whole. Most struck idiophones are percussive in nature, i.e., the sonorous instrument is struck with a non-sonorous object, for example with a wooden stick or the human hand. Since this is the type of instrument we aim to produce, studying their shapes might give us hints on how to choose our design space.

Common representatives of struck idiophones in western culture are the triangle, the marimba, the xylophone, the glockenspiel, the vibraphone, the cymbal, the steel pan, the wood block, the cow bell, and the bell.

The glockenspiel is an example of how a high range of pitches can be reached by using simple shapes, namely rectangular metal plates, and varying only a single shape parameter, the length of plate. However, the overtones of the glockenspiel are not tuned in any way because a single shape parameter does not provide a sufficiently large design space. The marimba resembles the glockenspiel in that it also consists of a set of oblong bars, although usually made from rosewood. The big difference is that marimba bars are additionally tuned by adding arched undercuts to the bottom. This way, a harmonic

ratio of 1:4 between the pitch and the first overtone can be achieved in order to produce dulcet tones.

Cymbals resemble thin circular plates with a small dome in the center. They are usually made out of bronze, and their pitch is controlled by the diameter and the thickness of the plate. The steel pan is an instrument that started evolving in the mid-20th century. It is made from a circular steel plate that is bulged with a special 16 oz. hammer. The bulges are distributed throughout the plate such that nearby bulges give tones that are harmonically related. This leads to reciprocal excitations and a rich spectrum of sounds.

Bells are among the oldest and most widely distributed instruments. The weight of a bell can vary between several grams and more than a hundred tons. There are appreciable differences in shape, but most variations are radially symmetric and have either cylindrical or pear shapes. The pitch of most bells is roughly proportional to the thickness and inversely proportional to the squared diameter. The overtones of the sound are shaped by thinning the material in select regions on the inside of the bell. By this process, the first overtone is shaped to lie one octave above the pitch, and the second overtone a minor third above the first.

What all of these instruments have in common is that they are composed of thin plates or thin shells. A single parameter, e.g., the thickness, the height, or the diameter, suffices to dramatically change the pitch by at least an octave. Further variations in shape are either decorative or serve the improvement of sound quality and of the overtone spectrum.

Motivated by this common property of struck idiophones, we choose a thin-shell solid as our primary design. A thin-shell solid can be thought of as the volume enclosed between an outer surface with boundary, an inner surface with boundary, and the surface connecting the two boundaries. The distance between the two surfaces corresponds to the thickness of the shell. Using this description, we can reproduce many of the instruments described above. If the surfaces are two stacked rectangles, this describes a bar of a glockenspiel or a xylophone. If the surfaces resemble nested hemispheres, this allows us to describe bell-like resonators. Note that instead of using two surfaces with boundaries and connecting them, it is also possible to use two surfaces *without* boundaries to begin with, e.g., two nested spheres describing a spherical shell. Even though this type of shape is less common in the design of idiophones, it does not pose a problem for our optimization pipeline. In the case of bells, the outer surface is the defining aesthetic component. The inner surface, on the other hand, can be used to control the sound without compromising the visual appeal of the design.

Taking these considerations into account, we can define the input parameters and the design space as follows: The user is asked to input the frequency of the desired pitch, the parameters of the target material (material density ρ , Young's Modulus E , and Poisson's ratio ν for isotropic materials), and a triangle mesh describing the desired outer surface. The design space is parametrized by the shell thickness, which controls the distance between the inner surface and the user-defined outer surface. The inner surface is calculated as an offset surface to the outer surface.

We introduce notation to make the notion of the *inner surface*, which is created by



Figure 4.1: The mesh we use to demonstrate the three parametrizations of the design space.

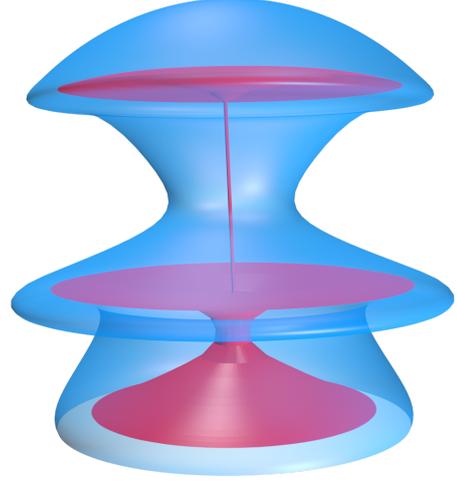


Figure 4.2: The meso-skeleton calculated for skeleton parametrization consists of surface sheets and curves.

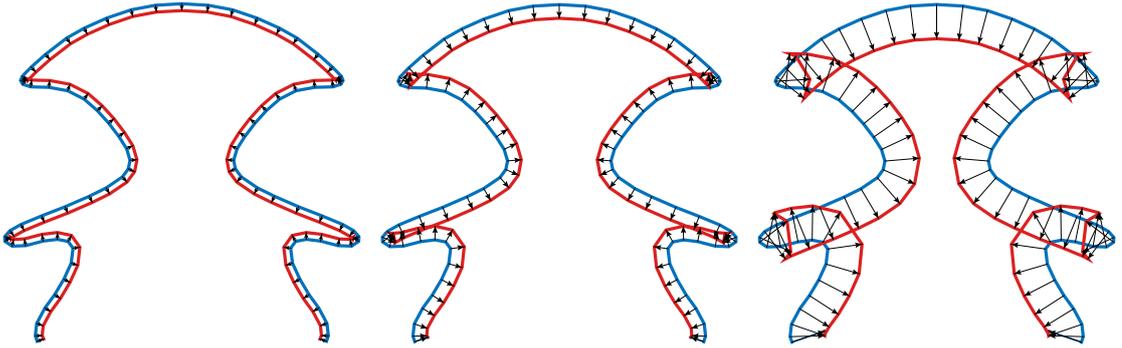


Figure 4.3: Inner surface calculated using per-vertex-normal parametrization. Even for small offsets, the inner surface exhibits self-intersections.

shifting the vertices of the outer surface along an offset direction, more concrete. Let \mathcal{O} denote the constant outer surface and $\mathbf{v}_1, \dots, \mathbf{v}_n$ the vertex positions of the triangle mesh describing \mathcal{O} . Let $\mathbf{o}_1, \dots, \mathbf{o}_n$ be a set of normalized offset directions corresponding to the vertices of \mathcal{O} . The offset magnitudes are denoted by $\delta_1, \dots, \delta_n$. The vertex positions of the inner surface \mathcal{I} are given by $\mathbf{v}_i + \delta_i \mathbf{o}_i$. The vertex-edge-face topology of the inner surface is defined to be the same as the outer surface.

What is left to choose are the offset directions \mathbf{o}_i and how to parametrize the offset magnitudes δ_i . We have explored three possibilities to address this challenge and describe each of them in turn.

4.2.2 Per-Vertex-Normal Parametrization

The simplest choice for the offset directions is to use the inward-facing per-vertex normals \mathbf{n}_i . The design space is parametrized by a single variable $\delta > 0$ such that $\delta_i = \delta$ for all vertices. This results in an almost constant wall thickness, i.e., a constant distance between the two surfaces.

There is an obvious problem with this formulation: Depending on the shape of the outer surface and the value of δ , the inner surface could intersect either itself or the outer surface. Self-intersections can be either local or global. Local self-intersections occur when the offset magnitude at a point is higher than the inverse of the maximal curvature κ_1 . In the case of smooth surfaces, this is easy to see because the inner offset surface will develop a singularity if the offset magnitude matches or exceeds the curvature radius. Predicting local self-intersections for discrete surfaces is less reliable because the notion of curvature is only defined in an approximate sense.

The situation is even more difficult for global self-intersections, i.e., one region of the inner surface intersects another region that is not directly bordering the first. The only way to reliably detect these intersections is to perform triangle-triangle tests. The same holds true for intersections between the inner and the outer surface. The implication is that it is difficult to find an upper bound for δ in advance. It is equally difficult to find per-vertex offset magnitude upper bounds $\bar{\delta}_i$. We demonstrate per-vertex-normal parametrization for the mesh shown in Fig. 4.1. Fig. 4.3 shows the inner surface at different distances to the original surface in a 2d illustration. In regions of high curvature, the method fails even for small offsets.

Nevertheless, this simple parametrization of the design space is useful for outer surfaces with low curvatures. It can be successfully applied to flat instruments and shapes like the hemiellipsoid bell discussed in Section 5.2.1.

4.2.3 Skeleton Parametrization

To address the intersection problems of the per-vertex-normal parametrization, we follow the lead of Musialski et al. [MAB⁺15]. The idea is to compute a meso-skeleton, an approximation of the medial axis for triangle meshes. The medial axis of a surface is the set of all centers of spheres that are tangent to the surface in at least two points. Equivalently, it is the set of all points that are equidistant to the surface in at least two points. In three dimensions, the medial axis consists of curves and surface sheets.

Tagliasacchi et al. [TAOZ12] propose an algorithm that produces a skeleton, i.e., a set of curves, in the end but yields a meso-skeleton as an intermediate step by iterative mesh contraction. A useful property of their algorithm is that the one-to-one matching between a vertex \mathbf{m}_i of the meso-skeleton and a vertex \mathbf{v}_i of the initial surface is preserved. The idea is to choose the offset directions as

$$\mathbf{o}_i = \frac{\mathbf{m}_i - \mathbf{v}_i}{\|\mathbf{m}_i - \mathbf{v}_i\|}, \quad 0 < \delta_i < \|\mathbf{m}_i - \mathbf{v}_i\|.$$

This means that a vertex of the inner surface lies on the line segment connecting the corresponding vertex \mathbf{v}_i on the outer surface and the corresponding vertex \mathbf{m}_i on the

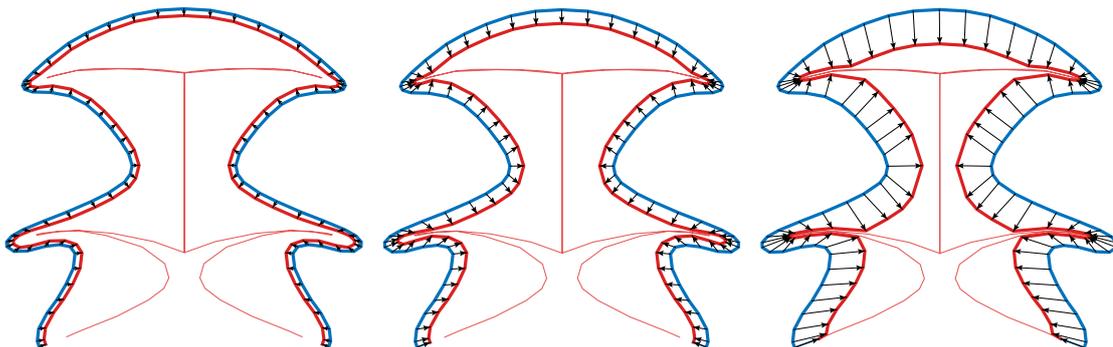


Figure 4.4: Inner surface calculated using skeleton parametrization. The inner surface avoids local and global self-intersections as well as intersections with the outer surface.

meso-skeleton. As the wall thickness δ_i is increased, the vertex is shifted closer to the skeleton. The effect of skeleton parametrization, as opposed to per-vertex-normal parametrization, is illustrated in Fig. 4.4.

Like the previous parametrization method, skeleton parametrization also uses a single design-space parameter δ , which controls the distance between the two surfaces. However, the offset directions \mathbf{o}_i no longer coincide with the per-vertex normals \mathbf{n}_i in general. Therefore, setting $\delta_i = \delta$ will not actually achieve a wall thickness of δ but a reduced wall thickness of

$$t_i = \delta_i \mathbf{o}_i^T \mathbf{n}_i.$$

Often it is useful to enforce a minimum wall thickness w throughout the object due to stability concerns or restrictions of the fabrication process. The approximate wall thickness t_i at vertex i can be found by projecting the offset vector $\delta_i \mathbf{o}_i$ onto the normal: To enforce that δ_i respects the minimum wall thickness, we can introduce the lower-bound constraints

$$\underline{\delta}_i < \delta_i, \quad \underline{\delta}_i = \frac{w}{\mathbf{o}_i^T \mathbf{n}_i}.$$

Additionally, we modify the upper-bound constraints

$$\delta_i < \bar{\delta}_i, \quad \bar{\delta}_i = c \|\mathbf{m}_i - \mathbf{v}_i\|,$$

where c is a constant that is close to but less than one. This constant factor guarantees that there is always a small gap between opposite regions of the inner surface. In our implementation we set $c = 0.95$.

The meso-skeleton approach works well in practice, but it is not unconditionally robust. It can happen that the meso-skeleton does not extend tendrils into very fine surface details—“jags”. As a consequence, the line segment connecting \mathbf{v}_i and \mathbf{m}_i may intersect the outer surface or the trajectory of other surface points. This means that self-intersections can still occur for very noisy surfaces.

A practical detail that should be mentioned is that mean-curvature skeletons are only defined for 2-manifolds without boundary. If the outer surface \mathcal{O} , which is part

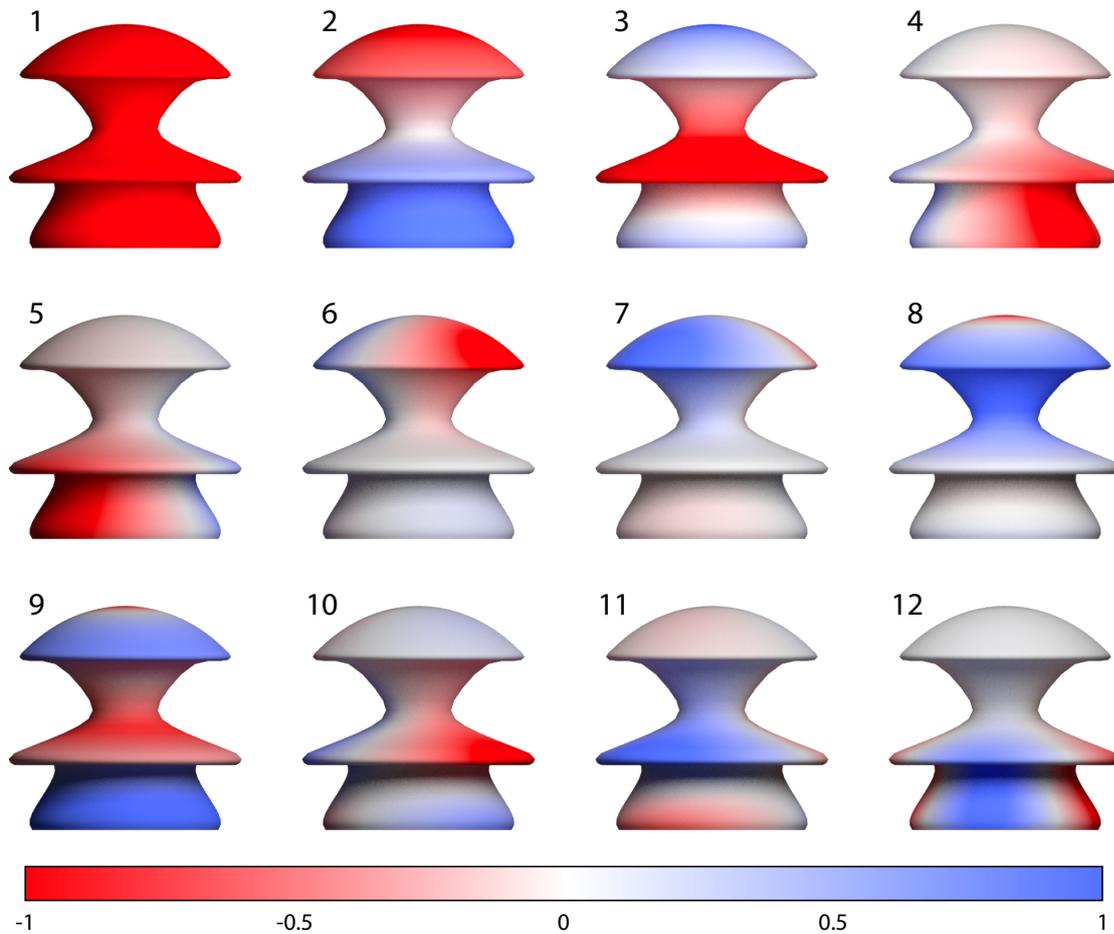


Figure 4.5: The first twelve Laplacian eigenfunctions of a mesh. The color encodes the influence regions of the individual functions. The first eigenfunction is always constant.

of the user input, has a boundary, we cannot directly feed it into the skeletonization algorithm. Instead we ask the user to input an extended closed surface $\tilde{\mathcal{O}}$ such that $\mathcal{O} \subset \tilde{\mathcal{O}}$. The surface $\tilde{\mathcal{O}}$ has annotations that identify the faces pertaining to \mathcal{O} . In the case of a hemiellipsoidal bell, the user could input the entire ellipsoid and annotate only one half as belonging to \mathcal{O} . The surface $\tilde{\mathcal{O}}$ is then used to build the meso-skeleton, and the vertices not contained in \mathcal{O} are discarded afterwards.

4.2.4 Manifold Harmonics Parametrization

If the design space afforded by a single thickness parameter δ is insufficiently large, we can use a technique introduced by Musialski et al. [MAB⁺15] to extend the number of parameters. This might be necessary if additional design goals are to be met, e.g., the optimization of overtones or aesthetic considerations. The order-reduction method presented here strikes a balance between a single design parameter δ and n per-vertex

design parameters. This is achieved by projecting the set of offset magnitudes δ_i into a lower-dimensional space.

The projection is based on the mesh Laplacian, a symmetric linear operator that encodes the mesh topology and mesh geometry. It is defined via the matrix \mathbf{L} with entries

$$L_{i,j} = \begin{cases} w_{i,j} & \text{if } (i,j) \in \mathcal{E} \\ -\sum_k w_{i,k} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases},$$

where \mathcal{E} denotes the set of edges of the mesh. The vertex weights are computed as

$$w_{i,j} = \frac{1}{2}(\cot \varphi_{i,j}^l + \cot \varphi_{i,j}^r),$$

where $\varphi_{i,j}^l$ and $\varphi_{i,j}^r$ are the angles opposite the edge (i,j) .

Applying the discrete Laplacian operator to the 3-by- n matrix of vertex coordinates $\mathbf{V} = [\mathbf{x}|\mathbf{y}|\mathbf{z}]$ transforms them to δ -coordinates, i.e., the vectors pointing from the initial vertex positions to the weighted mean of their 1-neighbors. The Laplacian operator has applications in mesh encoding and mesh compression due to the nature of its eigenbasis. Because the Laplacian is real symmetric, its eigendecomposition can be written as

$$\mathbf{L} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T,$$

where $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ is the diagonal matrix of eigenvalues $\lambda_1 < \dots < \lambda_n$, and $\mathbf{Q} = [\mathbf{q}_1|\mathbf{q}_2|\dots|\mathbf{q}_n]$ is the orthogonal matrix whose columns are the eigenvectors of \mathbf{L} . The eigenvectors of the Laplacian correspond to the eigenfunctions of the Laplace-Beltrami operator in differential geometry.

The discrete mesh Laplacian can be seen as a generalization of the discrete Fourier Transform to arbitrary mesh topologies, where the eigenvectors take on the meaning of the Fourier basis vectors. They share the property that eigenvectors corresponding to small eigenvalues encode low frequency features of the mesh and eigenvectors corresponding to large eigenvalues encode high-frequency features of the mesh. Fig. 4.5 illustrates the first twelve eigenvectors of a mesh by interpreting them as scalar functions on the surface of the mesh. Truncation of the series

$$\mathbf{L} = \sum_{i=1}^n \lambda_i \mathbf{q}_i \mathbf{q}_i^T$$

to the $k < n$ lowest-frequency terms

$$\mathbf{L}_k = \sum_{i=1}^k \lambda_i \mathbf{q}_i \mathbf{q}_i^T = \mathbf{Q}_k \mathbf{\Lambda}_k \mathbf{Q}_k^T$$

yields a reduced spectral basis that encodes only low-frequency surface features.

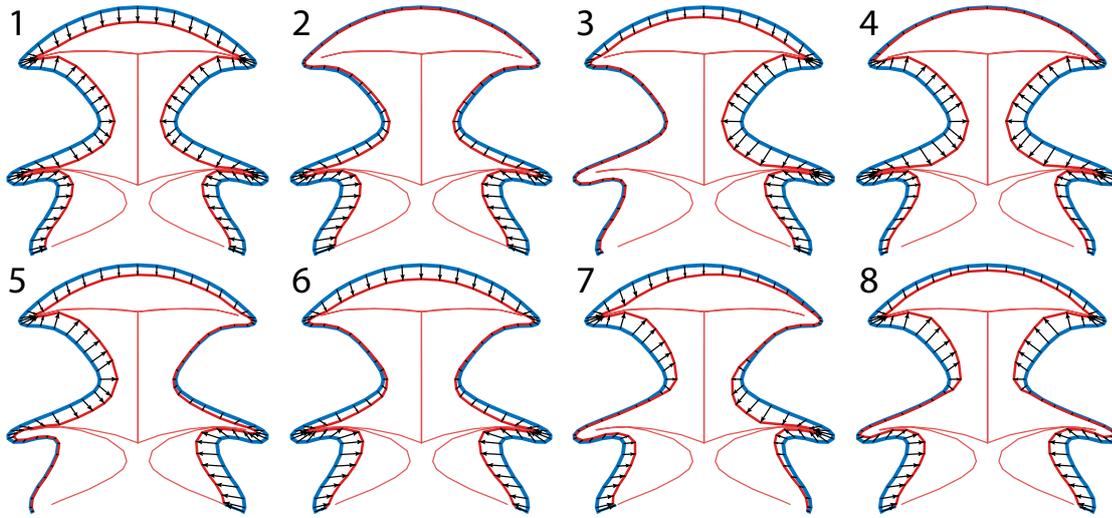


Figure 4.6: Manifold harmonics parametrization. Each Laplacian eigenfunction changes the wall thickness in a different region of influence.

As proposed by Musialski et al. [MAB⁺15], we do not transform the mesh vertices themselves by projection on the reduced eigenbasis. Instead, we use the k lowest-frequency eigenvectors to parametrize the offset magnitudes $\boldsymbol{\delta} = (\delta_1, \dots, \delta_n)^T$ as a linear combination of the eigenvectors with coefficients $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_k)$

$$\boldsymbol{\delta} = \sum_{i=1}^k \alpha_i \mathbf{q}_i.$$

The coefficients $\boldsymbol{\alpha}$ serve as a the new set of parameters. Fig. 4.6 shows a 2d illustration of each summand $\alpha_i \mathbf{q}_i$ for $i \in \{1, \dots, 8\}$. By linearly combining the offsets, a large design space with smoothly varying wall thickness can be constructed.

The first eigenvector \mathbf{q}_1 is always equal to $(1, \dots, 1)^T$. This means that the manifold harmonics parametrization degenerates into the skeleton parametrization method from Section 4.2.3 if k is chosen to be one. In practice it makes sense to keep k as small as possible (e.g., $k < 4$) because optimization run-times increase drastically as the number of parameters increases.

No matter which of the three parametrization methods is used, an element of the design space always has the same form: an outer triangle mesh and an inner triangle mesh, both of which have the same mesh topology. If one of the two latter methods is used and the constraints on δ_i are respected, the inner surface will not intersect either itself or the outer surface with a high likelihood.

4.3 Finite Element Modeling

Given an element of the design space, this section describes how to calculate its natural frequency spectrum. A design consists of an outer surface and an inner offset surface, both of which are stored as triangle meshes with the same topology. If the two surfaces are closed, we can define the domain $\Omega \subset \mathbb{R}^3$ as the region delimited by the two surfaces. In the case of surfaces with boundary, the domain is defined by first connecting the boundaries of the outer and the inner surface. This yields one closed 2-manifold $\partial\Omega$ that encloses Ω .

Section 2.3 describes the principal ideas behind a finite element model. The most important choice when attempting a finite element analysis is that of the element types to use. The linear tetrahedron is the most conceptually simple element, but more complex elements have higher predictive power for complex structural problems. We have implemented modal analysis with four different types of elements and will describe each of them in turn.

4.3.1 Linear Tetrahedra

The element introduced in Section 2.3 spans the convex hull of four corner nodes. At each node, there are three degrees of freedom in the form of a displacement vector \mathbf{u} , totaling twelve degrees of freedom per element. The first step is to construct an assemblage that partitions Ω into a number of tetrahedral elements Ω^e .

By connecting a triangle of the outer surfaces with the corresponding triangle of the inner surface, we can construct a triangular prism. Doing this for every pair of triangles yields a partition of Ω into a set of triangular prisms. As illustrated in Fig. 4.7 on the left, a triangular prism can be subdivided into three tetrahedra, but there is a choice between six distinct configurations that can be obtained by rotational and reflection symmetry. If we were to choose a configuration at random, we will run into trouble regarding the quads that join two prisms together. It must be ensured that the quads are split into triangles in the same direction on both neighboring elements. The reason is that the interpolation of nodal quantities must be continuous across element boundaries in order to guarantee correctness of the finite element method. If two triangulated quads are joined together in cross fashion, as seen in Fig. 4.7 on the right, this will no longer be the case.

Dompierre et al. describe an algorithm to tetrahedralize a mesh comprised of triangular prisms without introducing non-matching elements [DLVC99]. The algorithm is based on a global numbering of all vertices in the mesh. Each prism contains three quadrangular faces, and each one is split along the vertex with the lowest global number. It is shown that this always yields a valid tetrahedralization of the prism, and hence a globally matching tetrahedral mesh.

Section 2.3 describes how to derive the element stiffness matrix and the element mass matrix for the standard linear tetrahedron spanned by the vertices $\{(1, 0, 0)^T, (0, 1, 0)^T, (0, 0, 1)^T, (0, 0, 0)^T\}$. In order to derive the element matrices for elements with arbitrary nodal positions, a coordinate transformation is required. We introduce the notion of the natural coordinate system $\boldsymbol{\xi} = (\xi, \eta, \zeta)^T$. In contrast to the global coordinate system

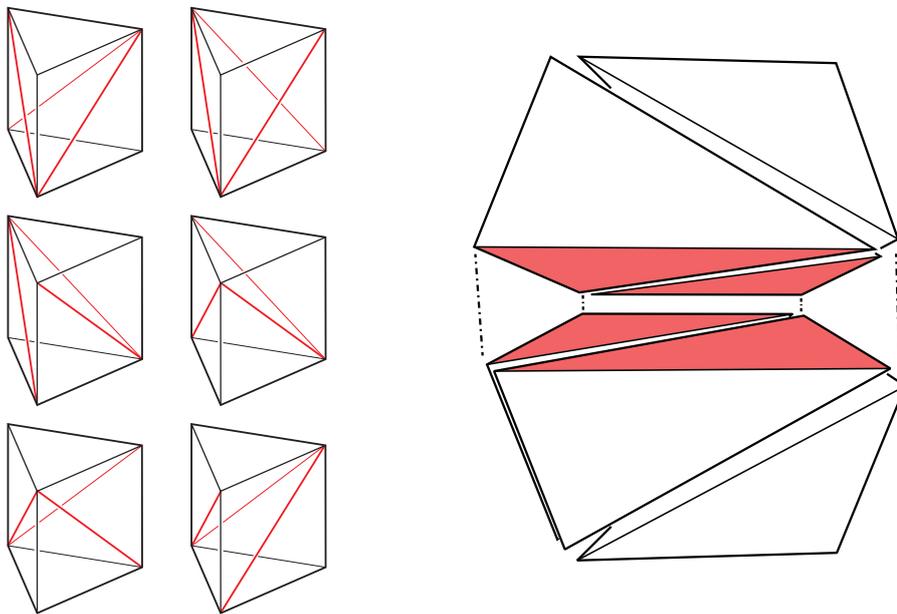


Figure 4.7: Left: The six possible ways to subdivide a triangular prism into three tetrahedra. Right: Joining two prisms in cross fashion results in a non-matching finite element mesh.

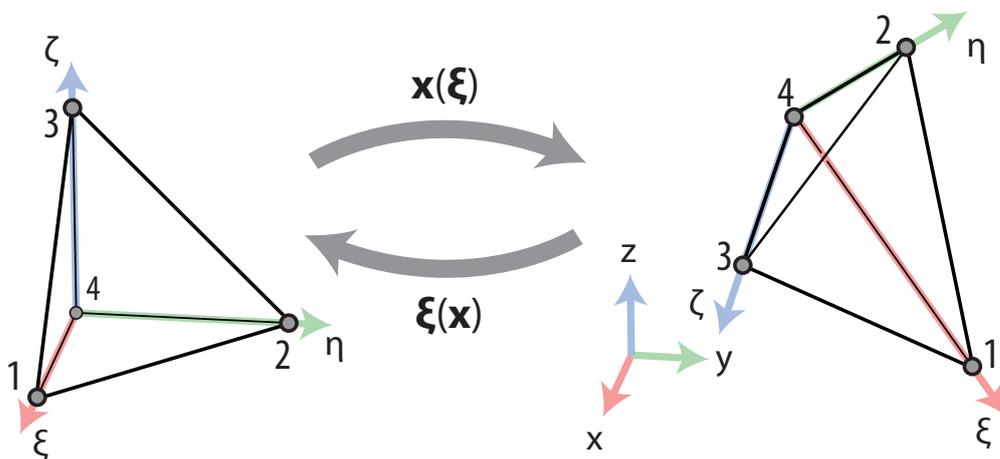


Figure 4.8: A linear tetrahedral element in the natural coordinate system (ξ, η, ζ) and in the global coordinate system (x, y, z) .

$\mathbf{x} = (x, y, z)^T$, which is the same for each element Ω^e , the natural coordinate system is chosen locally for each element. It is constructed such that any element maps to the standard tetrahedral element Ω^0 in the natural coordinate system, see Fig. 4.8.

Given a point $\boldsymbol{\xi} = (\xi, \eta, \zeta)^T \in \Omega^0$ in natural coordinates, it can be mapped to the corresponding point $\mathbf{x} = (x, y, z)^T \in \Omega^e$ using the familiar shape functions

$$\begin{aligned} N_1^e(\boldsymbol{\xi}) &= \xi, \\ N_2^e(\boldsymbol{\xi}) &= \eta, \\ N_3^e(\boldsymbol{\xi}) &= \zeta, \\ N_4^e(\boldsymbol{\xi}) &= 1 - \xi - \eta - \zeta \end{aligned}$$

with the mapping

$$\mathbf{x}(\boldsymbol{\xi}) = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{pmatrix} \begin{pmatrix} N_1^e(\boldsymbol{\xi}) \\ N_2^e(\boldsymbol{\xi}) \\ N_3^e(\boldsymbol{\xi}) \\ N_4^e(\boldsymbol{\xi}) \end{pmatrix},$$

where $(x_i, y_i, z_i)^T$ denotes the position of node i . Note that positions in the global coordinate system are interpolated in precisely the same way as other nodal quantities like the displacements

$$\mathbf{u}(\boldsymbol{\xi}) = \sum_{i=1}^4 N_i^e(\boldsymbol{\xi}) \mathbf{u}_i.$$

Elements that show this analogy between geometry and other nodal quantities are known as isoparametric, as opposed to superparametric elements, whose geometry is less refined than its unknowns.

In order to perform a transformation between the global and the natural coordinate system, we make use of the Jacobi matrix

$$\mathbf{J} = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{pmatrix}$$

and its inverse

$$\frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}} = \mathbf{J}^{-1}.$$

In the case of a linear tetrahedral element, \mathbf{J} and \mathbf{J}^{-1} are constant across the whole element because they depend only on the constant derivatives of the shape functions. This does not hold for more complicated elements, where \mathbf{J} and \mathbf{J}^{-1} may depend on $\boldsymbol{\xi}$.

The entries of \mathbf{J}^{-1} enable the derivation of \mathbf{B} , whose definition is repeated here for convenience:

$$\bar{\mathbf{e}}(\mathbf{x}) = \begin{pmatrix} \frac{\partial N_1^e}{\partial x} & 0 & 0 & \dots & \frac{\partial N_4^e}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_1^e}{\partial y} & 0 & \dots & 0 & \frac{\partial N_4^e}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_1^e}{\partial z} & \dots & 0 & 0 & \frac{\partial N_4^e}{\partial z} \\ \frac{\partial N_1^e}{\partial y} & \frac{\partial N_1^e}{\partial x} & 0 & \dots & \frac{\partial N_4^e}{\partial y} & \frac{\partial N_4^e}{\partial x} & 0 \\ 0 & \frac{\partial N_1^e}{\partial z} & \frac{\partial N_1^e}{\partial y} & \dots & 0 & \frac{\partial N_4^e}{\partial z} & \frac{\partial N_4^e}{\partial y} \\ \frac{\partial N_1^e}{\partial z} & 0 & \frac{\partial N_1^e}{\partial x} & \dots & \frac{\partial N_4^e}{\partial z} & 0 & \frac{\partial N_4^e}{\partial x} \end{pmatrix} \mathbf{u}^e = \mathbf{B}\mathbf{u}^e.$$

Since our shape functions depend on the natural coordinates $\boldsymbol{\xi}$, we cannot calculate the entries of \mathbf{B} directly. Instead, we apply the chain rule, which yields

$$\frac{\partial N_1^e}{\partial x} = \frac{\partial N_1^e}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial N_1^e}{\partial \eta} \frac{\partial \eta}{\partial x} + \frac{\partial N_1^e}{\partial \zeta} \frac{\partial \zeta}{\partial x}$$

and analogous expressions for the remaining entries of \mathbf{B} . Using matrix calculus notation, they can be written as

$$\frac{\partial \mathbf{N}^e}{\partial \mathbf{x}} = \frac{\partial \mathbf{N}^e}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}} = \frac{\partial \mathbf{N}^e}{\partial \boldsymbol{\xi}} \mathbf{J}^{-1},$$

where $\mathbf{N}^e = (N_1^e, N_2^e, N_3^e, N_4^e)^T$.

We are now in a position to evaluate the element stiffness matrix

$$\mathbf{K}^e = \int_{\Omega^e} \mathbf{B}^T \mathbf{E} \mathbf{B} dV.$$

Because the integrand is constant, this simplifies to

$$\mathbf{K}^e = V \mathbf{B}^T \mathbf{E} \mathbf{B},$$

where V is the volume of the tetrahedral element

$$V = \frac{1}{6} \begin{vmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \\ 1 & 1 & 1 & 1 \end{vmatrix}.$$

The element mass matrix

$$\mathbf{M}^e = \int_{\Omega^e} \rho \mathbf{N}^T \mathbf{N} dV$$

for a linear tetrahedral element has the form

$$\mathbf{M}^e = \int_{\Omega^e} \rho \left(\begin{array}{ccc|ccc|ccc} N_1^2 & & & N_1 N_2 & & & \cdots & N_1 N_4 & & & \\ & N_1^2 & & & N_1 N_2 & & \cdots & & N_1 N_4 & & \\ & & N_1^2 & & & N_1 N_2 & \cdots & & & N_1 N_4 & \\ \hline N_1 N_2 & & & N_2^2 & & & \cdots & N_2 N_4 & & & \\ & N_1 N_2 & & & N_2^2 & & \cdots & & N_2 N_4 & & \\ & & N_1 N_2 & & & N_2^2 & \cdots & & & N_2 N_4 & \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \\ \hline N_1 N_4 & & & N_2 N_4 & & & \cdots & N_4^2 & & & \\ & N_1 N_4 & & & N_2 N_4 & & \cdots & & N_4^2 & & \\ & & N_1 N_4 & & & N_2 N_4 & \cdots & & & N_4^2 & \\ \hline & & & & & & \cdots & & & & N_4^2 \end{array} \right) dV,$$

where the subscript e has been omitted from N_i^e . Because every entry in the integrand is a sum of polynomials in ξ , η , ζ , and $(1 - \xi - \eta - \zeta)$, it can be integrated analytically with the general formula

$$\int_{\Omega^e} \xi^i \eta^j \zeta^k (1 - \xi - \eta - \zeta)^l dV = \frac{i! j! k! l!}{(i + j + k + l + 3)!} 6V.$$

for tetrahedral integration domains Ω^e . For the two kinds of entries appearing in \mathbf{M}^e , this yields the expressions

$$\begin{aligned} \int_{\Omega^e} \rho N_i(\boldsymbol{\xi})^2 dV &= \rho \int_{\Omega^e} \xi^2 dV = \frac{1}{10} \rho V, \\ \int_{\Omega^e} \rho N_i(\boldsymbol{\xi}) N_j(\boldsymbol{\xi}) dV &= \rho \int_{\Omega^e} \xi \eta dV = \frac{1}{20} \rho V, \quad i \neq j. \end{aligned}$$

The construction of both \mathbf{K}^e and \mathbf{M}^e is very efficient for linear tetrahedra because no numerical integration has to be performed. This makes the linear tetrahedron an attractive choice for prototyping purposes.

Unfortunately, this element does not perform well in the context of modal analysis, and the frequencies predicted by the finite element model do not converge to the true solution even if the mesh resolution is increased. This phenomenon is known as shear locking and appears if elements of insufficient continuity are used to predict bending of a structure. In the case of linear tetrahedra, only displacements possess C^0 continuity across element boundaries. Stresses and strains on the other hand are constant on the interior of an element and exhibit large jumps across element boundaries. In order to predict bending phenomena, stresses must be interpolated at least linearly in the element interiors.

4.3.2 Quadratic Tetrahedra

To address the shortcomings of linear tetrahedra, one can introduce additional nodes on the edges of the elements, called mid-side nodes. This increases the number of nodes per element from four to ten. Fig. 4.9 illustrates the placement and numbering of a

quadratic tetrahedron. The most notable difference to the linear tetrahedron is that the new element supports curved boundaries. This also means that its natural coordinate system is curvilinear.

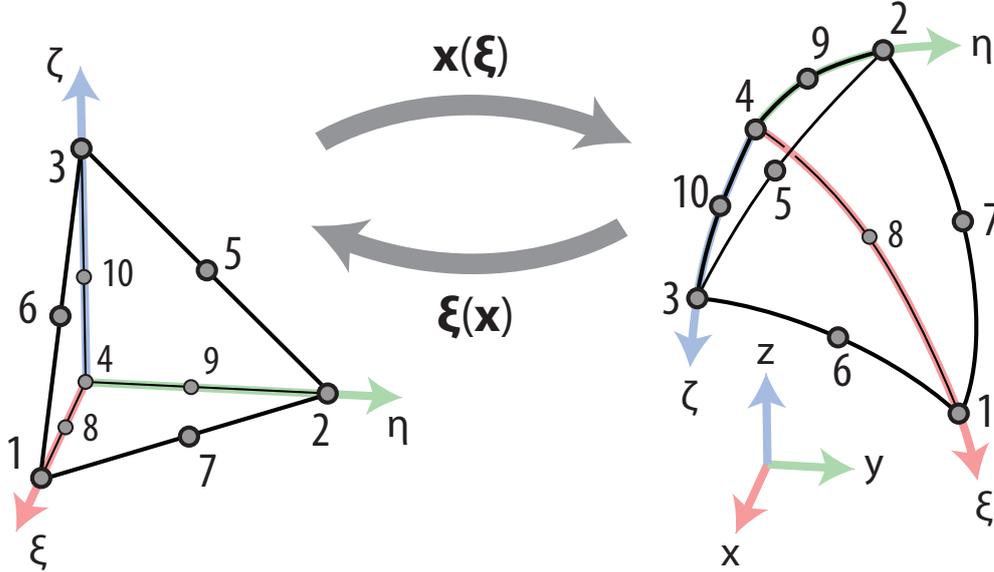


Figure 4.9: A quadratic tetrahedral element in the natural coordinate system (ξ, η, ζ) and in the global coordinate system (x, y, z) .

To define interpolation on a quadratic tetrahedron, it suffices to provide a set of shape functions N_1^e, \dots, N_{10}^e . In constructing these shape functions, we have to keep the following criteria in mind:

- The shape functions must be a partition of unity on Ω^e .
- Each shape function must evaluate to one at the corresponding node and to zero at any other node.

It is easiest to construct the shape functions on the standard element in the natural coordinate system. For corner nodes like node 3 we can meet both criteria if we construct the equations of the planes shown in Fig. 4.10 on the left. Taking their product and scaling it to be one at the position of node 3 gives us a suitable shape function [Fel14]:

$$\begin{aligned} \varepsilon_1 : \zeta = 0, \quad \varepsilon_2 : \zeta - \frac{1}{2} = 0, \\ N_3^e(\boldsymbol{\xi}) = 2\zeta \left(\zeta - \frac{1}{2} \right). \end{aligned}$$

We can do a similar construction for mid-side nodes like node 5.

$$\begin{aligned} \varepsilon_1 : \zeta = 0, \quad \varepsilon_3 : \eta = 0, \\ N_5^e(\boldsymbol{\xi}) = 4\eta\zeta. \end{aligned}$$

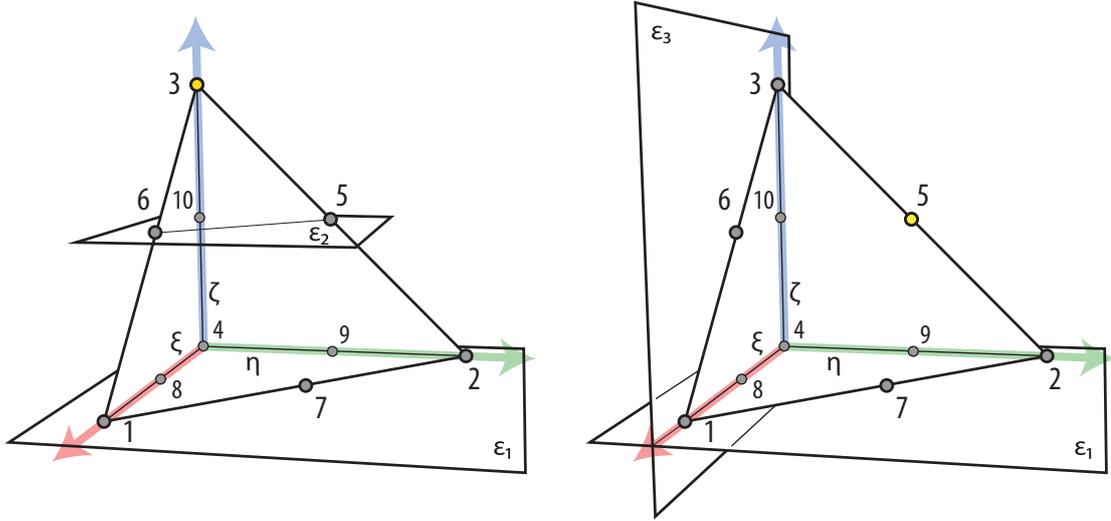


Figure 4.10: Planes for the construction of the shape functions of nodes 3 (left) and 5 (right) for the quadratic tetrahedron..

The eight remaining shape functions can be constructed analogously.

With the shape functions in place, we can readily evaluate \mathbf{B} for any point $\boldsymbol{\xi}$. A complication for quadratic elements is that we can no longer evaluate the integral in the definition of \mathbf{K}^e analytically. Instead, we perform a coordinate transformation

$$\mathbf{K}^e = \int_{\Omega^e} \mathbf{B}^T \mathbf{E} \mathbf{B} dV = \int_{\Omega^0} \mathbf{B}^T \mathbf{E} \mathbf{B} |J| dV, \quad J = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}},$$

into the natural coordinate system, where $|J|$ denotes the Jacobi determinant.

This integral over the domain of the unit tetrahedron can be numerically approximated by Gauss quadrature with the formula

$$\mathbf{K}^e \approx \sum_{i=1}^{n_G} w_i \mathbf{B}^T(\mathbf{g}_i) \mathbf{E} \mathbf{B}(\mathbf{g}_i) |J(\mathbf{g}_i)|,$$

where n_G is the number of quadrature points, w_i is the weight of quadrature point i , and \mathbf{g}_i are the natural coordinates of quadrature point i . It is suitable to choose the simplest integration rule that will produce a rank-sufficient stiffness matrix. The rank of \mathbf{K}^e is 24 because the quadratic tetrahedral element has 30 degrees of freedom and six rigid-body modes. The rank of the integrand $\mathbf{B}^T \mathbf{E} \mathbf{B}$ is six, given that E has full rank. Each summand in the quadrature rule adds six to the rank of the resulting stiffness matrix, and therefore at least four quadrature points are necessary to achieve a rank of 24. Since a 4-point integration rule is available for tetrahedra, it is a suitable choice for the integration of the element stiffness matrix.

The construction of the mass matrix is also performed by substituting coordinates in the integral

$$\mathbf{M}^e = \int_{\Omega^e} \rho \mathbf{N}^T \mathbf{N} dV = \int_{\Omega^0} \rho \mathbf{N}^T \mathbf{N} |J| dV$$

and evaluated using Gauss quadrature

$$\mathbf{M}^e \approx \sum_{i=1}^{n_G} w_i \rho \mathbf{N}(\mathbf{g}_i)^T \mathbf{N}(\mathbf{g}_i) |J(\mathbf{g}_i)|,$$

however, the 4-point rule will not produce a rank-sufficient matrix. Every summand adds only three to the rank of the resulting matrix, and therefore at least ten points are required to achieve the full rank of 30. Since a symmetric 10-point rule for tetrahedra is not available, the best choice is 14-point Gauss quadrature.

4.3.3 Quadratic Wedges

While quadratic tetrahedral elements converge to the true result, finite element literature recommends to use prismatic elements over tetrahedra if possible. Our volume initially consists of triangular prisms, and it is therefore viable to use 15-node quadratic wedge elements. We construct these by introducing a mid-side node on every edge of the triangular prism, as shown in Fig. 4.11 on the top-left.

This element is particularly elegant for our scenario because it captures the geometry of the solid perfectly without the need to subdivide elements or to introduce nodes that do not add meaningful degrees of freedom. It might even be possible to reduce the wedge element to 12 nodes by removing the mid-side nodes 10-12. We include these nodes because the FEM library that we use to construct the stiffness and mass matrices does not support the 12-node wedge element.

The quadratic wedge element has three different types of nodes: corner nodes (nodes 1-6), base mid-side nodes (nodes 7-9 and 13-15), and axial mid-side nodes (nodes 10-12). Using the product-of-planes construction method, we can come up with shape functions for each type of node, as illustrated in Fig. 4.11. The corresponding shape functions are given by

$$\varepsilon_1 : 1 + \xi = 0, \quad \varepsilon_2 : \zeta = 0, \quad \varepsilon_3 : \frac{1}{2}\xi + \zeta - 1 = 0,$$

$$N_5^e(\boldsymbol{\xi}) = (1 + \xi)\zeta \left(\frac{1}{2}\xi + \zeta - 1 \right),$$

$$\varepsilon_1 : 1 + \xi = 0, \quad \varepsilon_2 : \zeta = 0, \quad \varepsilon_4 : 1 - \xi = 0,$$

$$N_{11}^e(\boldsymbol{\xi}) = (1 - \xi^2)\zeta,$$

$$\varepsilon_1 : 1 + \xi = 0, \quad \varepsilon_2 : \zeta = 0, \quad \varepsilon_5 : \eta = 0,$$

$$N_{13}^e(\boldsymbol{\xi}) = 2(1 + \xi)\zeta\eta.$$

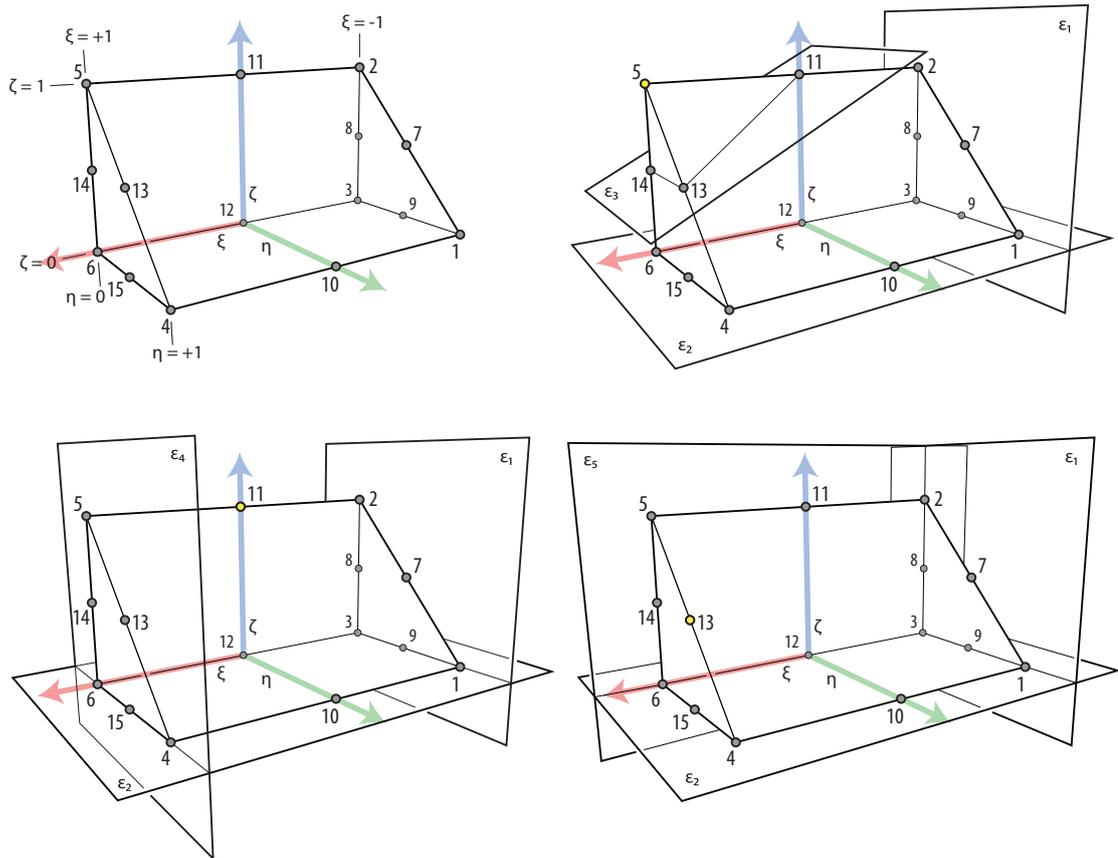


Figure 4.11: Top-left: The 15-node wedge element with dimension markings. Remaining: Planes for the construction of the shape functions of nodes 5 (top-right), 11 (bottom-left), and 13 (bottom-right).

The remaining elements can be constructed in the same way, and the coordinate transformations are performed in analogy to the calculation of quadratic tetrahedral elements.

Gauss quadrature can be performed by choosing a triangle quadrature rule for the (η, ζ) -plane and a one-dimensional quadrature rule for the ξ -direction. The 15 nodes have 45 degrees of freedom in total, which yields a rank of 39 for the element stiffness matrix. Again, each summand of the form $\mathbf{B}^T \mathbf{E} \mathbf{B}$ adds six to the rank of the result, and therefore at least seven quadrature points are required. An appropriate choice is a 4-point triangle rule and a 2-point one-dimensional rule, which yields an 8-point rule for the wedge-element stiffness matrix.

The element mass matrix has a rank of 45, and each summand of the form $\mathbf{N}^T \mathbf{N}$ adds three to the rank of the result. Therefore, we need at least 15 quadrature points. A valid choice is a 6-point triangle rule and a 3-point one-dimensional rule, which yields an 18-point rule for the wedge element mass matrix.

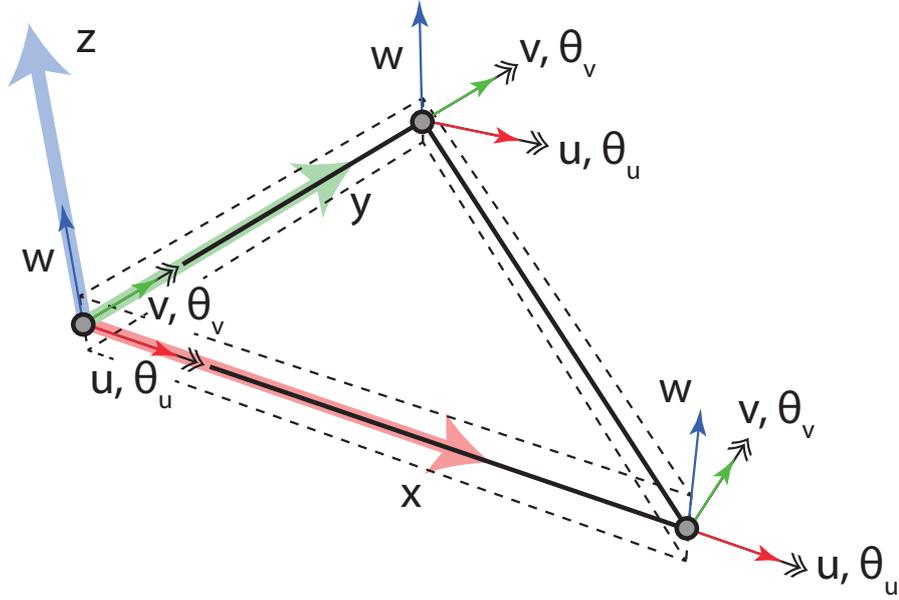


Figure 4.12: A triangular thin-shell element with five degrees of freedom at each node—the membrane degrees of freedom u, v and the thin-plate degrees of freedoms w, θ_u, θ_v .

4.3.4 Thin-Shell Elements

Thin-shell structures are composed of curved surfaces whose thickness is considerably smaller than its longitudinal dimensions. An assemblage of thin-shell elements can be used to describe these structures with a smaller number of nodes and degrees of freedom than are necessary with the quadratic elements introduced before. Unlike linear tetrahedral elements, thin-shell elements can predict tensile stresses as well as bending stresses.

The reduction of DoF is made possible by distributing all nodes along the mid-surface of the thin-shell structure. In contrast to the elements explored previously, there are no nodes on the outer and inner surfaces. The transverse dimension takes on an implicit role by including the thickness as a parameter in the energy formulation of the element.

An element suitable for our application is the triangular thin-shell element. Its nodes are placed on the mid-surface of a triangular prism, see Fig. 4.12. Every node has five degrees of freedom—two displacements to account for in-plane vibrations, and an additional translational and two rotational degrees of freedom to describe flexural vibrations. Derivation of the stiffness matrix for this element is a little more involved than the one described in Section 2.3. The interested reader is referred to a finite element textbook [Pet10].

The thin-shell formulation only works well as long as the distance between the outer surface and the inner surface is small. However, we cannot always guarantee that the distance stays small throughout the optimization routine at every vertex. If we expect large wall thicknesses, we use quadratic wedge elements instead as they provide a good

compromise between correctness and ease of calculation.

4.3.5 Jacobian Positiveness

With the introduction of the Jacobian $|J|$, we run the risk of violating the vital assumptions that \mathbf{K}^e and \mathbf{M}^e are positive semi-definite. Take as a simple example the element stiffness matrix of the linear tetrahedron, in which the integrand is constant. The integral can be solved by multiplying the integrand with the volume of the integration domain. However, if the orientation of the domain Ω^e is flipped with respect to the standard linear tetrahedron Ω^0 , the volume will be negative. As a result, the stiffness matrix of this element will be negative semi-definitive, and the positive semi-definiteness of the master stiffness matrix \mathbf{K} is no longer guaranteed.

For elements whose shape functions are at least bilinear or quadratic, like the quadratic tetrahedron, the situation is more complicated. Since the Jacobian varies across the integration domain, it can be positive in one region, negative in another, and zero in between. The Jacobi matrix \mathbf{J} is only evaluated at a finite set of Gauss integration points, but if one of these points has a zero Jacobian, the inversion of \mathbf{J} will fail. If the Jacobian is negative, the element stiffness matrix can still be calculated, but the element does not have a valid physical shape. Therefore, most finite element codes will issue a warning if the Jacobian is not positive everywhere.

Section 4.2 describes how we avoid self-intersections of the inner surface and intersections between the inner and the outer surface. Local self-intersections of the inner surface automatically lead to inverted elements and negative Jacobians. By avoiding intersections, we remove the main cause for invalid element shapes. In extreme cases, it is possible to arrive at severely distorted, concave elements with negative Jacobians even in the absence of self-intersections. However, we have never encountered these cases in practice when using skeleton parametrization or manifold harmonics parametrization.

4.3.6 Remarks

A major part of the run-time of the shape optimization pipeline is spent on solving the equations of motion. The complexity of this operation depends mostly on the size of the master matrices, i.e., the total number of degrees of freedom in the finite element mesh. Therefore, we evaluate the number of elements and total number of degrees of freedom that we can expect when choosing an element type.

An infinite regular triangle mesh that is extruded to form a single layer of triangular prisms serves as the basis of our considerations because it has a topology similar to that of a finely tessellated mesh. As a measure of complexity, we evaluate the ratio between the number of prisms and the total number of degrees of freedom in the finite element model. Fig. 4.13 illustrates the nodes resulting from different element types.

For the thin-shell model, there is one element per prism, three axial mid-side nodes (green) per element, five DoF at each node, and each node is shared by six elements. This yields $3 \cdot 5 \cdot \frac{1}{6} = 2.5$ DoF per prism in the limit.

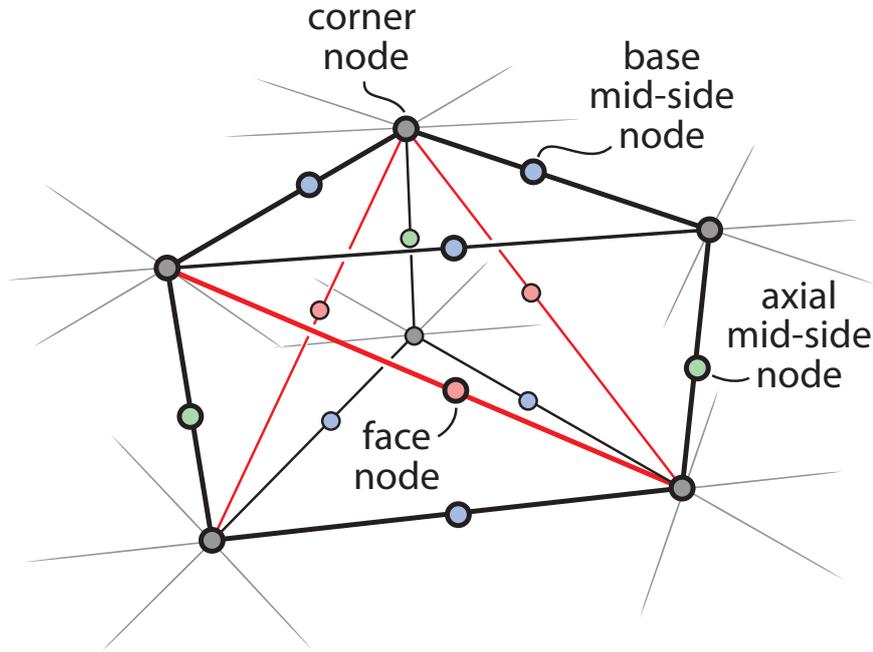


Figure 4.13: A prism in an infinite regular mesh is decomposed into quadratic tetrahedra. The different types of nodes are marked in different colors.

For the wedge element, we have to distinguish between the different types of nodes. An element has six corner nodes (gray) and three axial mid-side nodes (green) that are shared by six elements each. Furthermore, it has six base mid-side nodes (blue) that are shared by two elements each. Given three DoF per node, this yields $3 \cdot \left((6 + 3) \cdot \frac{1}{6} + 6 \cdot \frac{1}{2} \right) = 13.5$ DoF per prism.

For the tetrahedral elements, every prism is subdivided into three tetrahedra with the scheme introduced in Section 4.3.1. The six corner nodes (gray) are shared by six prisms each. Given three DoF, we thus have $3 \cdot 6 \cdot \frac{1}{6} = 3$ DoF per prism for the linear tetrahedron. For the quadratic tetrahedron, we have all nodes that were present for the wedge element plus three face nodes (red), which are shared by two prisms. This yields a total of $3 \cdot \left((6 + 3) \cdot \frac{1}{6} + (3 + 6) \cdot \frac{1}{2} \right) = 18$ DoF per prism.

We can conclude that the master finite element matrices obtained with the thin-shell model or the linear tetrahedral model are far smaller than the ones obtained with the quadratic models. Since linear tetrahedra are not suitable for our application, we expect the thin-shell model to have the best run-time performance. However, tests with the finite element library OpenFEM have shown that construction of the thin-shell element matrices is comparatively slow. In some of our test cases, this problem outweighs the disadvantageously large master matrices of the wedge model. The quadratic tetrahedral model is by far the slowest of the four models for every test case.

4.4 Non-Linear Optimization

Given an element of the design space—an outer surface and an inner offset surface—we can evaluate the natural frequency spectrum using any of the finite element discretizations introduced in the previous section. The stiffness matrix and the mass matrix can be fed into a generalized eigenproblem solver for sparse matrices in order to find the smallest eigenvalues with the corresponding eigenvectors. As discussed in Section 2.4, the first eigenvalue is always zero and has a multiplicity of six, corresponding to the three translational and three rotational rigid-body modes of a 3d solid. The seventh eigenvalue is positive and describes the frequency of the pitch. This task of determining the first seven eigenvalues is delegated to Matlab’s function `eigs`, which implements the Lanczos algorithm to iteratively find eigenvalues.

We proceed by defining a global optimization problem, first for the case of per-vertex-normal parametrization:

$$\begin{aligned} \min_{\delta} \quad & f(\delta) = (p - p_0)^2, \\ \text{s.t.} \quad & \delta > 0. \end{aligned}$$

The objective function f is the squared difference between the desired pitch p_0 and pitch of the incumbent solution p . The wall thickness parameter δ has to be positive to keep the inner surface inside the outer surface. The individual offset magnitudes are set to $\delta_i = \delta$ for every vertex.

The inequality constraint is linear, but the objective function involves the eigenvalue of a matrix and is therefore non-linear. Evaluating the objective function is very expensive because it requires creating the inner surface based on δ , generating the stiffness matrix and the mass matrix, and finding the smallest non-zero eigenvalue λ_7 of a large, sparse generalized eigenproblem. The frequency of the pitch is then found to be $p = \frac{\sqrt{\lambda_7}}{2\pi}$.

To solve the minimization problem, we use Matlab’s `fmincon`, which implements constrained non-linear optimization algorithms. We have experimented with both the active set algorithm and the SQP algorithm and found that the latter outperforms the former in all of our test cases.

If we use skeleton parametrization, the offset magnitudes δ_i are set to δ but clamped to the individual offset limits $\underline{\delta}_i$ and $\bar{\delta}_i$ to guarantee a minimum wall thickness and a safe distance to the skeleton. Let $\delta_{\min} = \min_i \underline{\delta}_i$ and $\delta_{\max} = \max_i \bar{\delta}_i$. Then we can write the optimization problem for skeleton parametrization as

$$\begin{aligned} \min_{\delta} \quad & f(\delta) = (p - p_0)^2, \\ \text{s.t.} \quad & \delta_{\min} < \delta < \delta_{\max}. \end{aligned}$$

Whenever the objective function is evaluated, the individual offset magnitudes are set to

$$\delta_i = \min(\max(\delta, \underline{\delta}_i), \bar{\delta}_i).$$

Manifold harmonics parametrization increases the number of optimization variables from 1 to k , depending on the number of basis functions. The variable δ is superseded

by the variables $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_k)^T$, which determine the weights of the individual eigenvectors $\mathbf{q}_1, \dots, \mathbf{q}_k$. The offset magnitudes are given by

$$\boldsymbol{\delta} = \sum_{i=1}^k \alpha_i \mathbf{q}_i.$$

The optimization problem is adapted to reflect this change

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & f(\boldsymbol{\alpha}) = (p - p_0)^2, \\ \text{s.t.} \quad & \underline{\boldsymbol{\delta}}_i \preceq \mathbf{Q}_k \boldsymbol{\alpha} \preceq \bar{\boldsymbol{\delta}}_i, \end{aligned}$$

where $\mathbf{Q}_k = [\mathbf{q}_1 | \mathbf{q}_2 | \dots | \mathbf{q}_k]$, $\underline{\boldsymbol{\delta}}_i = (\underline{\delta}_1, \dots, \underline{\delta}_n)^T$ and $\bar{\boldsymbol{\delta}}_i = (\bar{\delta}_1, \dots, \bar{\delta}_n)^T$.

As opposed to the original application of offset-surface parametrization [MAB⁺15], we cannot analytically compute the gradient and the Hessian of the optimization function. Therefore, the SQP routine uses finite differencing methods to find said quantities, which increases the number of objective function evaluations. If we use more than one optimization variable, this number will increase even further. Given the time it takes to evaluate the objective function, it is ill-advised to choose k larger than four.

The optimization task also poses a numerical difficulty. Although the objective function $f(\delta)$ is monotonic on large intervals for many of our test cases, the numerical inaccuracies in evaluating the eigenvalues with `eigs` introduce very small-scale jags into the function. To the SQP algorithm, these jags appear to be local minima, and they are often returned instead of a feasible solution. This problem mostly occurs if the change in variables in order to determine the gradient by finite differencing is very small, e.g., δ is changed by less than 10^{-6} .

The minimum and maximum change in variables can be set for `fmincon` with the parameters `DiffMinChange` and `DiffMaxChange`. While the outcome is largely insensitive to the value of `DiffMaxChange`, it can change dramatically depending on the value of `DiffMinChange`. If the minimum change in variables is very small, e.g. 10^{-6} , the algorithm often returns a local minimum that only exists due to the inaccuracies of `eigs`. If the minimum change in variables is larger, e.g. 10^{-3} , the algorithm approaches the true minimum, but returns a value that is up to about 2% off.

An approach that works well in practice is a two-phase strategy. The optimization routine is run once with a large minimum change in variables to obtain a good estimate. The routine is run again using the estimate as an initial solution and with a small minimum change in variables to refine the solution.

Validation and Results

5.1 Validation Method

This section describes the method we use to assess the quality of the finite element solutions and the results of the optimization pipeline. The most important metric is the difference between the frequencies predicted by the FE solution and the frequencies actually exhibited by the fabricated models.

A natural mode in structural dynamics is defined by three quantities: a frequency, a damping factor, and a mode shape. In the finite element framework described in Section 2.3, we use an undamped model to describe natural modes. This eliminates the damping factor from the list of parameters in which we are interested. The frequency and the mode shape emerge naturally from the solution of the linear system of second-order ODEs as the eigenvalues and eigenvectors of the corresponding generalized eigenvalue problem.

Modal analysis can be performed on a physical object with a series of experiments: For lightweight structures it is common to use a combination of an impact hammer and a set of accelerometers in order to perform modal testing. The accelerometers are attached to nodes throughout the test subject and will record the acceleration history. The impact hammer is equipped with a force cell at its tip to measure the force and duration of an impact. All components are hooked up to modal testing software, which will process the experimental data.

The theoretical ideal impulse is the so-called Dirac impulse, which has an infinitely small duration. This impulse would excite all frequencies equally and therefore completely characterize an ideal linear dynamical system. Since it is impossible to recreate in reality, the impact delivered with the impact hammer is recorded and transformed into the frequency spectrum by the modal testing software. This input spectrum determines the excited frequency range and therefore the part of the output spectrum that will contain valid data. The recorded accelerations are also transformed into frequencies and

correlated with the input spectrum. From this data, the modal testing software estimates the natural frequencies, damping factors, and even mode shapes.

5.1.1 Experimental Frequency Analysis

The only experimental data we require to assess the quality of the finite element solution are the frequencies of the natural modes in the audible spectrum, i.e., 20 to 20,000 Hz. Therefore, we forgo the standard experimental modal analysis setup in favor of a more economic validation method. The fabricated solid is mounted between loops of yarn to simulate free-free boundary conditions. Ideal free-free boundary conditions would require that the object is not supported at all, but only supporting it with strings is a good approximation [GPSG11]. It is then excited with a short impulse that is delivered with a sharp object, e.g., a screwdriver. We record the sound with a standard microphone and extract the frequency spectrum using the discrete short-time Fourier transform (STFT). This variation of the discrete Fourier transform characterizes the frequency spectrum of a signal as it changes over time. This is achieved by breaking up the signal into several short time segments and performing a Fourier transform on each one. We use Matlab's `spectrogram` function to perform STFT with overlapping windows and the Hamming window function. The result is a matrix \mathbf{P} of power spectral densities, where each row corresponds to a time segment and each column corresponds to a frequency between 0 and 22,000 Hz. The power spectral density is defined as the squared magnitude of the amplitude of a particular frequency. A logarithmic plot of the power spectral densities, with the frequency in Hz along the abscissa and the time in seconds along the ordinate, conveys the dominant frequencies as shown in Fig. 5.1

The function `spectrogram` internally uses FFT to compute the Fourier transform of each time segment. Therefore, the number of samples in each segment is padded with zeros to yield a number of elements that is power-of-two by default. An additional option allows for an artificially larger padding that leads to a higher resolution of the result along the frequency axis. While this does not increase the information content, it makes it easy to interpolate amplitudes between frequencies. This is a desirable quality because we are interested in the frequency peaks, which correspond to natural modes of the object.

In order to extract the frequency peaks, we opt for a two-phase method. The goal of the first phase is to identify a set of approximate peak locations, along with their approximate widths and durations. The second phase performs surface fitting for each peak in order to determine its precise location.

For the first phase, we have implemented two variants: a simple interactive method that lets the user select peaks, and an automatic method that detects probable peak locations. In the interactive variant, the user is asked to select a frequency range $[f_{\min}, f_{\max}]$ around a peak and its approximate extension along the time dimension $[0, t_{\max}]$ on the spectrogram.

The automatic variant searches for peaks on the second time segment from the beginning of the signal. The first time segment is often subject to noise, and therefore it is more robust to use the second. In the following description, p refers to power

spectral density, and s is the derived quantity $10 \cdot \log_{10}(p)$. The quantity s is used to plot spectrograms, and we find it easier to formulate the peak-finding algorithm on s rather than on p . We perform the following steps to find, narrow down, and characterize peak locations (f_i, s_i) :

1. Perform non-maximum suppression [Can86] on s to rule out anything that is not a local maximum.
2. Exclude any point that is smaller than $s_{\max} - 40$, where s_{\max} is the maximal value of s on the time segment.
3. From the remaining points, exclude points f_i that are not maximal in the frequency range $[f_i - 120, f_i + 120]$.
4. For each peak, determine the maximal region $[f_i - l_i, f_i]$ in which s is monotonically increasing, and the maximal region $[f_i, f_i + r_i]$ in which s is monotonically decreasing. Let $o_i = \min(l_i, r_i)$, then the range of the peak along the frequency axis is assumed to be $[f_{\min}, f_{\max}] = [f_i - \frac{9}{10}o_i, f_i + \frac{9}{10}o_i]$.
5. For each peak, define a threshold $t_i = \min(s_{\max} - 40, s_i - 30)$. Then the extension of a peak along the time axis is given by the maximal duration $[0, t_{\max}]$, such that s never falls below t_i during this time interval.

The peaks found by this algorithm sometimes include peaks that do not correspond to a natural mode and have to be removed manually. Fig. 5.2 illustrates the output of the peak finder, omitting the extension of the peaks along the time axis for clarity.

The second phase of the peak-extraction pipeline is also automatic and performs local optimization in order to fit a surface of the form

$$g : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad g : (f, t; A, D, \mu, \sigma) \mapsto Ae^{-Dt}e^{-\left(\frac{f-\mu}{\sigma}\right)^2}$$

to the selected range of power spectral densities with the Gauss–Newton algorithm. The function g represents a Gaussian bell curve $Ae^{-\left(\frac{f-\mu}{\sigma}\right)^2}$ along the frequency axis, which decays exponentially with the term e^{-Dt} along the time axis. The parameter A represents the height of the bell curve’s peak, μ is the location of the peak—this is what we are ultimately interested in—, σ is the standard deviation of the bell curve, and D controls the damping along the time axis.

Let n be the number of power spectral density samples in the range $[f_{\min}, f_{\max}] \times [0, t_{\max}]$ that corresponds to the region covered by a peak. We enumerate these samples as $(f_1, t_1, p_1), \dots, (f_n, t_n, p_n)$, where f_i is the frequency, t_i is the time, and p_i is the power spectral density of a sample. The goal of the Gauss–Newton algorithm is to minimize the residuals

$$r_i = g(f_i, t_i) - p_i, \quad i \in \{1, \dots, n\}$$

in a sum of squares sense with respect to the parameters A , D , μ , and σ .

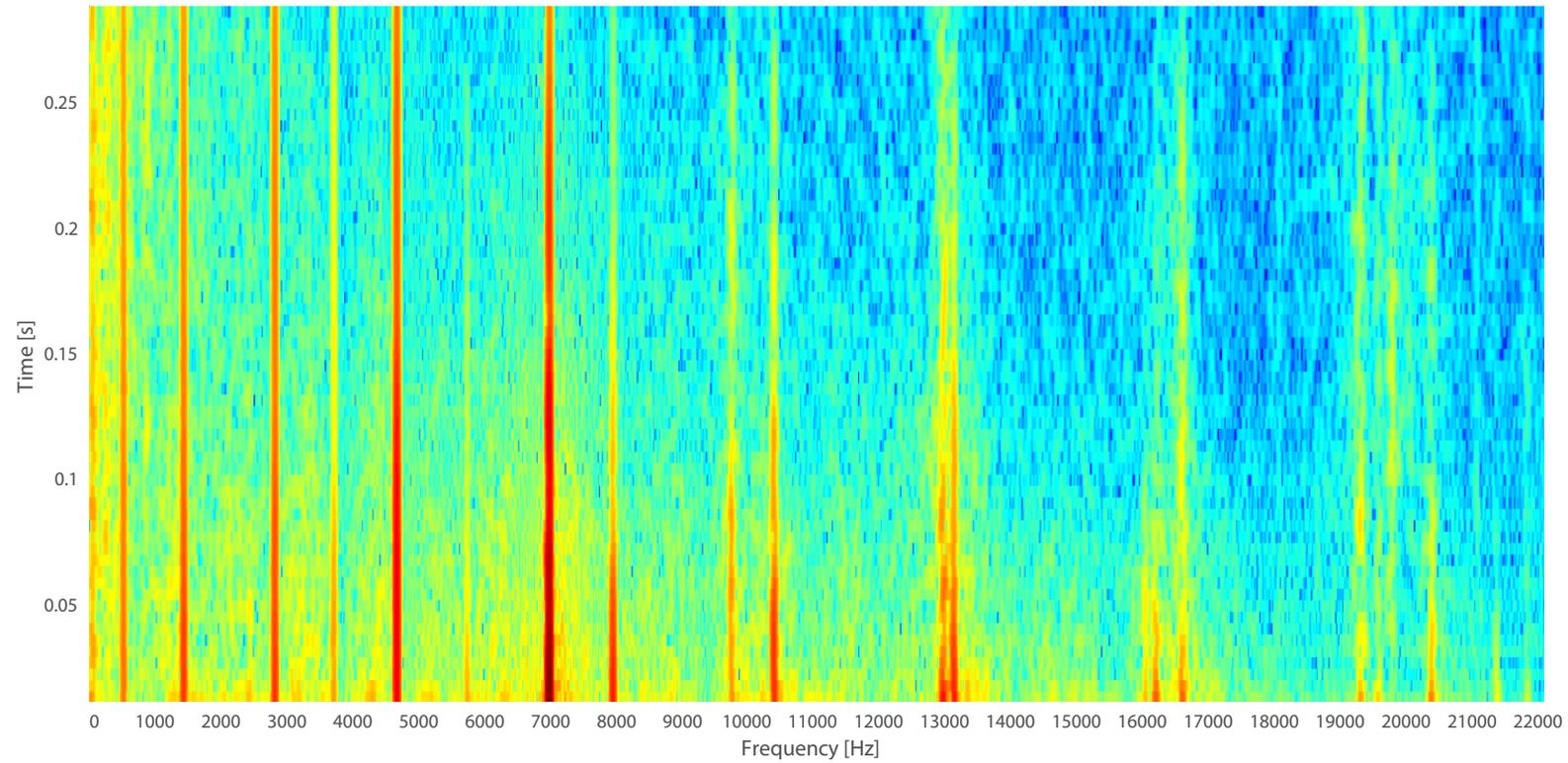


Figure 5.1: The power spectrogram of a sound sample for plate 3. Red regions mark high spectral powers, blue regions mark low spectral powers.

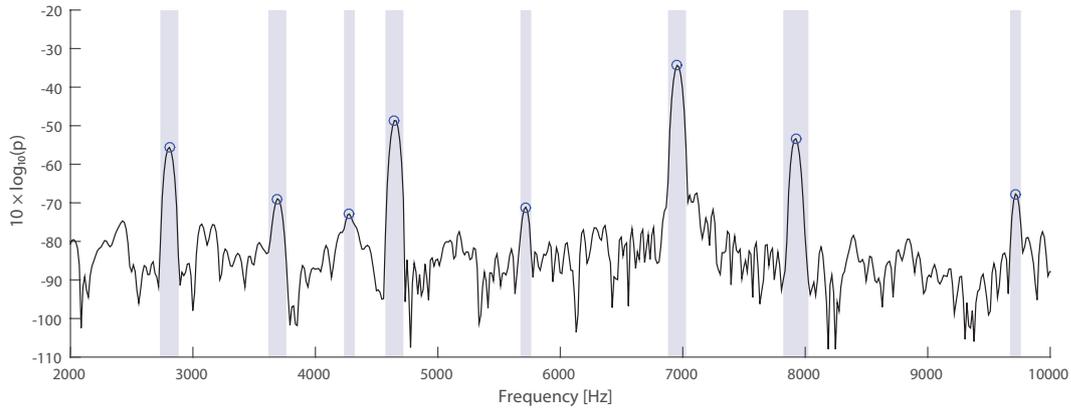


Figure 5.2: This figure shows part of the time segment of the spectrogram on which peak finding is performed. Blue circles mark estimated peak locations. The blue, shaded regions mark the estimated frequency range of a peak. The third peak from the left is found by the automatic peak finder although it does not actually correspond to a natural mode and has to be removed manually.

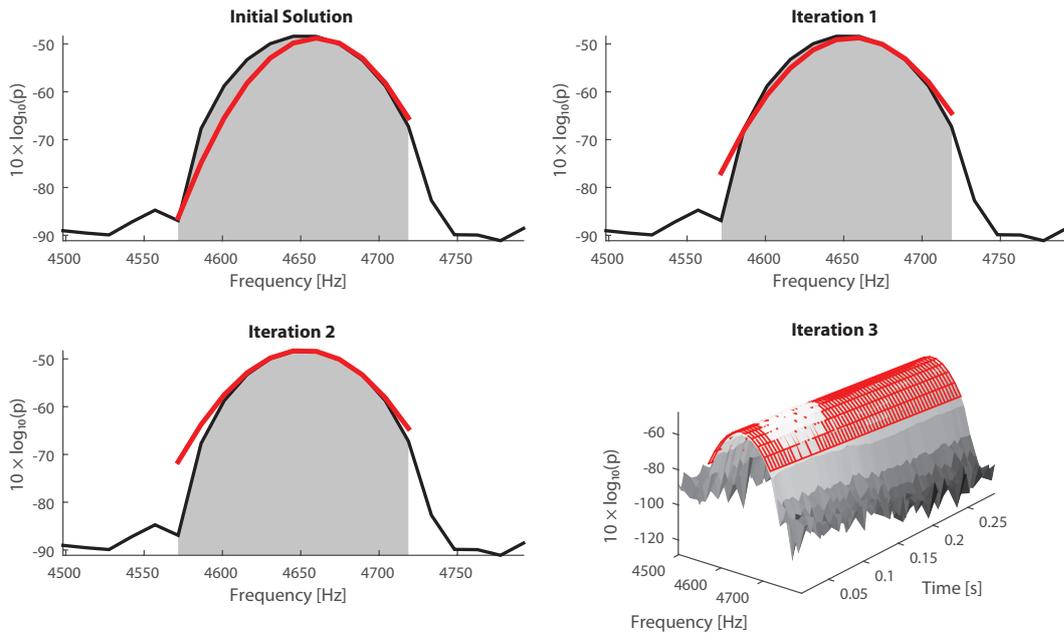


Figure 5.3: The 2d diagrams show a part of a time segment that contains a peak. The gray, shaded area marks the interval that is being fitted with a surface. The time axis has been omitted for clarity. After two Gauss–Newton iterations, a good fit has been obtained. Note that fitting is performed on the power spectral densities p , but the ordinate shows $10 \cdot \log_{10}(p)$. Therefore, the deviations between the two functions in the lower regions are negligible—a good fit in the upper part of the function is important. The 3d diagram includes the time axis to show the entire fitted surface.

Using the Gauss–Newton algorithm requires the Jacobi matrix of the residuals with respect to the estimation parameters, which is given by

$$\mathbf{J}_r = \frac{\partial \mathbf{r}}{\partial \boldsymbol{\beta}} = \begin{pmatrix} \frac{\partial r_1}{\partial A} & \frac{\partial r_1}{\partial D} & \frac{\partial r_1}{\partial \mu} & \frac{\partial r_1}{\partial \sigma} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial r_n}{\partial A} & \frac{\partial r_n}{\partial D} & \frac{\partial r_n}{\partial \mu} & \frac{\partial r_n}{\partial \sigma} \end{pmatrix}.$$

The individual entries can be expressed as

$$\begin{aligned} \frac{\partial r_i}{\partial A} &= e^{-Dt_i} e^{-\left(\frac{f_i - \mu}{\sigma}\right)^2}, \\ \frac{\partial r_i}{\partial D} &= -DA e^{-Dt_i} e^{-\left(\frac{f_i - \mu}{\sigma}\right)^2}, \\ \frac{\partial r_i}{\partial \mu} &= 2A \frac{f_i - \mu}{\sigma^2} e^{-Dt_i} e^{-\left(\frac{f_i - \mu}{\sigma}\right)^2}, \\ \frac{\partial r_i}{\partial \sigma} &= 2A \frac{(f_i - \mu)^2}{\sigma^3} e^{-Dt_i} e^{-\left(\frac{f_i - \mu}{\sigma}\right)^2}. \end{aligned}$$

This minimization problem requires an initial solution

$$\boldsymbol{\beta}^{(0)} = (A_0, D_0, \mu_0, \sigma_0)^T,$$

but it turns out that the problem is largely insensitive to A_0 , D_0 , and σ_0 , and therefore we can choose them in a naive way. The important initial parameter μ_0 is set to $\frac{f_{\max} + f_{\min}}{2}$, the center of the region around the peak. The parameter A_0 is taken as the power spectral density of the sample $(\mu_0, 2)$, where 2 refers to the second row of P , i.e., the second time segment after the start of the signal. The first time segment is often subject to artifacts, and this is why we opt for the second. For D_0 and σ_0 we choose values of 1 and 30 respectively. These values are arbitrary choices of the same magnitude as the true damping and standard deviation values. Each Gauss–Newton iteration is performed with the formula

$$\boldsymbol{\beta}^{(s+1)} = \boldsymbol{\beta}^{(s)} - (\mathbf{J}_r^T \mathbf{J}_r)^{-1} \mathbf{J}_r^T \mathbf{r}(\boldsymbol{\beta}^{(s)}).$$

We observe convergence to five significant digits within four iterations on average. The surface fitting process is illustrated in Fig. 5.3.

The only parameter in which we are ultimately interested is the location of the peak along the frequency axis μ . The reason why we extend g along the time axis, and hence why we introduce the parameter D , is that it allows estimating μ from more samples.

So far we have explained how to estimate the frequency peaks from the sound samples of a single impact. However, it cannot be taken for granted that a single impact will excite every natural mode of the test subject, even if the impulse is a good approximation to the Dirac impulse. In particular, an impact mainly excites the natural modes whose mode shapes have a large displacement at the location of the impact. For example, take the first mode shape from Fig. 5.6: This mode will be excited strongly if the rectangular plate is hit in the center—the location of maximal displacement. If the bar is hit at a

quarter of its length, the blue region in Fig. 5.6, the mode will be excited less. For this reason, we extract the frequency peaks from impacts at various locations. However, our experiments show that the fundamental tone and the first few overtones can be extracted from any impact sound sample that we used.

Sometimes not every normal mode that is predicted by the finite element model can be found by our experimental modal analysis setup. The reason is that only flexural vibrations, i.e., vibrations along the normal of the surface, create large pressure waves in the air. Transverse vibrations, i.e., vibrations tangential to the surface, see mode 8 in Fig. 5.6, do not create large pressure waves and are sometimes inaudible. Another peculiarity of our setup is that low-frequency torsional modes, like mode 3 in Fig. 5.6, are suppressed because their maximal displacement is exactly where our test subject is supported by loops of yarn to keep it in suspension.

5.1.2 Material Parameter Estimation

A major factor in the accuracy of the finite element solution is the quality of the available material parameters: the material density ρ , Young's Modulus E , and Poisson's ratio ν . These parameters are usually determined experimentally, and listings for different alloys can be found, e.g., on MatWeb¹, a website dedicated to making data about material properties freely available. However, these properties may vary slightly depending on the producer and the production batch because they are sensitive to material purity and internal defects.

To determine the properties of a material with sufficient accuracy requires expensive equipment like a universal testing machine. As an alternative, we perform material parameter estimation from impact sound samples using optimization. To start with, we need a small set of physical artifacts with known dimensions that are produced from the material in question. For example, to estimate the parameter of the tin that we use in the production of bells, we produce two thin, rectangular tin plates. From these test subjects, we extract the frequency spectrum experimentally with the method outlined in Section 5.1.1. For the computational counterpart, we prepare finite element meshes that describe the test subjects and evaluate them using one of the finite element models discussed in Section 4.3. Initially, we use the expected material parameters obtained from data sheets.

These steps yield a set of experimentally determined frequencies m_i and a set of frequencies f_i obtained through finite element modal analysis. As discussed earlier, not every natural mode can be observed using our experimental setup, and therefore we cannot find a matching m_i for every f_i . However, the natural frequencies of our test subjects are spaced widely enough to make it possible to match pairs by hand. If a natural mode f_i does not have a matching m_i , it is ignored in the subsequent evaluation.

At this point we have obtained a set of pairs (m_i, f_i) , $i \in \{1, \dots, n\}$, where m_i denotes a measured frequencies and f_i denotes the corresponding frequency obtained with the FE model. Each pair corresponds to a natural mode of one the test subjects. In practice,

¹<http://www.matweb.com/>

we use low-frequency modes, e.g., the first six modes of each test subject, for material parameter estimation. Note that the f_i are functions of the material parameters ρ , E , and ν .

The only two parameters we estimate are ρ and E because ν is a well-documented parameter that does not usually deviate from the listed values. In order to estimate ρ and E , we formulate an optimization problem

$$\min_{\rho, E} f(\rho, E) = \sum_{i=1}^n \left| 1 - \frac{f_i(\rho, E)}{m_i} \right|.$$

We opt for the relative deviation measure $\left| 1 - \frac{f_i}{m_i} \right|$ instead of an absolute deviation measure like $|f_i - m_i|$ because it better corresponds to the sensitivity of human hearing.

This optimization problem suffers from the same numerical instability addressed in Section 4.4, i.e., the objective function contains small bumps due to the approximate eigenproblem solutions. Tuning the parameters `DiffMinChange` and `DiffMaxChange` does not work well with this formulation because the two variables ρ and E have greatly different magnitudes. Therefore, we scale the parameters in the following way. Let ρ_0 and E_0 denote the initial material property values from the data sheets. Then we introduce the dimensionless optimization parameters θ_ρ and θ_E to denote the relative deviation of the incumbent solution from ρ_0 and E_0 . Using these parameters, the optimization problem turns into

$$\min_{\theta_\rho, \theta_E} \hat{f}(\theta_\rho, \theta_E) = \sum_{i=1}^n \left| 1 - \frac{f_i(\theta_\rho \rho_0, \theta_E E_0)}{m_i} \right|. \quad (5.1)$$

We constrain both parameters θ_ρ and θ_E to the range $[0.8, 1.2]$ because we do not expect a deviation of more than 20% from the listed material parameters. Setting `DiffMinChange` to 10^{-4} and `DiffMaxChange` to 10^{-2} achieves robust optimization.

5.1.3 Aluminium Plate Benchmark

In order to assess the accuracy of the four finite element models discussed in Section 4.3, we use rectangular aluminium plates as benchmarks. The plates have been cut from a large aluminium plate with a thickness of 1.5 mm using a bandsaw. The listed material parameters for the aluminium plate are summarized in Table 5.2 in the column ‘‘Listed’’. The dimensions of the individual plates have been determined with a vernier scale to a precision of 0.1 mm and are summarized in Table 5.1.

For each of the differently sized plates, we define finite element meshes for the different element types. The basis for each mesh is a regular grid subdivision of the plate surface, such that every grid cell is approximately 3 by 3 mm wide. For example, for plate 2, which is 110.3 by 20.35 mm, this yields a 37-by-7 element grid. Every grid cell is split along the diagonal to yield a triangulation and extruded along the normal to achieve a height of 1.5 mm.

These steps yield a set of triangular prisms, which serve as the domains for wedge elements. For the finite element models using linear tetrahedral elements and quadratic

Plate ID	Length	Width	Height
0	50.75	20.35	1.5
1	81.15	20.25	1.5
2	110.3	20.35	1.5
3	121.6	20.35	1.5
4	150.7	20.3	1.5
5	181	20.3	1.5

Table 5.1: This table summarizes the dimensions of the rectangular aluminium plates used to benchmark the finite element models. All measurements are in millimeters.

Parameter	Aluminium	
	Listed	Optimized
Density ρ [kg/m ³]	2710	2774
Young's Modulus E [GPa]	68.0	66.4
Poisson's ratio ν	0.33	0.33

Table 5.2: Material properties for aluminium. The column “Listed” lists the parameters obtained from a data sheet, the column “Optimized” lists the parameters after parameter estimation.

tetrahedral elements, the triangular prisms are further subdivided into tetrahedra as described in Section 4.3.1. To use the thin-shell model, we use triangles along the mid-surface of the plate as element domains.

The natural frequencies computed with the four different finite element models can be found in Table 5.3 for plates 0 and 1, in Table 5.4 for plates 2 and 3, and in Table 5.5 for plates 4 and 5. These results have been obtained using the tabulated material parameters. The effects of using estimated material parameters are discussed further below. Additionally, these tables list the frequencies that have been experimentally determined by recording the impact sounds and performing the peak fitting method outlined above. The assignment of experimentally determined frequencies to the modes of the FE model was performed manually. The measured frequency field is left blank if the corresponding mode could not be observed in the sound samples, or if the assignment was ambiguous.

The cell colors reflect the relative error between the measured frequency and the frequency predicted by the finite element model. The color key is green for a deviation of 0%, yellow for a deviation of 3%, and red for a deviation of 6% or more.

It is immediately evident that the wedge model (W) yields the best results with almost every frequency below an error of 2.5%. The quadratic tetrahedral model (QT) yields similar results for low frequencies but slightly worse results for higher frequencies. The thin-shell model (TS) performs significantly worse for high frequencies with errors as high as 5%. For low frequencies however, TS yields results that are as good as W and QT. The results produced by the linear tetrahedral model (LT) are completely unusable,

with errors of up to 200%, and the LT model will therefore not be considered any further. The histograms and empirical cumulative distribution functions (CDF) of the relative frequency deviations for the models W, QT, and TS are plotted in Fig. 5.4. The plot reveals that the median error is 1.7% for the W model, 2.1% for the QT model, and 3.0% for the TS model.

In the next step, we apply the parameter estimation procedure to the first six modes of the W model. The optimized parameter values are listed in Table 5.2. The density and the Young's Modulus are adjusted by about 2.4% through the estimation procedure. Tables 5.6, 5.7, and 5.8 list the frequencies obtained with the optimized material parameters. It can be observed that the errors have been drastically reduced for the W, QT, and TS models, although only the W model has been used during parameter estimation. The empirical CDFs of the frequency deviations with the optimized parameters are plotted in Fig. 5.5. The median errors have been reduced to 0.7% for the W model, 0.5% for the QT model, and 0.9% for the TS model. From the tables we can see that mode 9 for plate 1, mode 8 for plate 2, modes 8 and 15 for plate 3, and mode 7 for plate 4 are outliers in that the prediction is significantly worse than for the neighboring modes. This is only true for model W and QT however; model TS does not show any outliers.

Fig. 5.6 shows the first 24 natural mode shapes of plate 3. Modes 8 and 15 represent in-plane vibrations of the plate, and those are exactly the two outliers for Table 5.7. Mode 3 is the lowest-frequency torsional mode and is not observed in our sound sample. The reason is that the plates are suspended by two loops of yarn at their endpoints, which is exactly where the mode shape has its maximum displacements.

The spectrogram of a sound sample for plate 3, from which the user selects the approximate locations of frequency peaks, is plotted in Fig. 5.1. The peaks extend far along the time axis, which means that the aluminium plate has low damping factors, especially for low-frequency modes. Because the aluminium plate produces a clear ringing sound, the natural modes are clearly separated from the background.

We draw two main conclusions from the results of the aluminium plate benchmark. Firstly, if the material parameters are chosen accordingly, the wedge model, the quadratic tetrahedral model, and the thin-shell model all yield pitch frequency predictions with a maximum error of 1.6%. This is a sizeable improvement on previous work [UMI10], in which the authors report that their fabricated flat plates have a relative frequency error of up to 6% with respect to the finite element solution before manually correcting the plates by filing corners and edges.

Secondly, comparing Figs. 5.4 and 5.5, we observe that the wedge model is influenced the least by the difference between the listed and the estimated parameter values, even though the wedge model itself was used to estimate the parameters. Since our fabrication method is crude, it is likely that we introduce defects into the material, e.g., slag from the melting furnace, grains of sand from the molds, or air bubbles, and therefore the wedge model might be preferable. This model has the added benefit that the triangular prismatic element domains emerge naturally from the offset-surface representation of the solid.

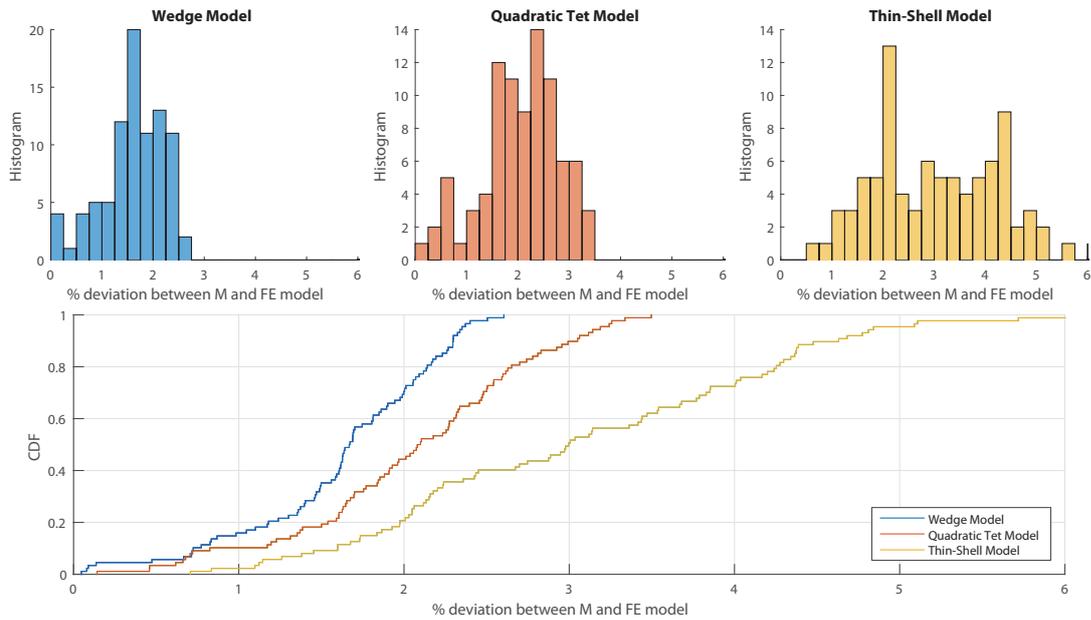


Figure 5.4: Histogram and empirical CDF of the relative deviations between the measured frequencies and the finite element model frequencies with the listed material parameters.

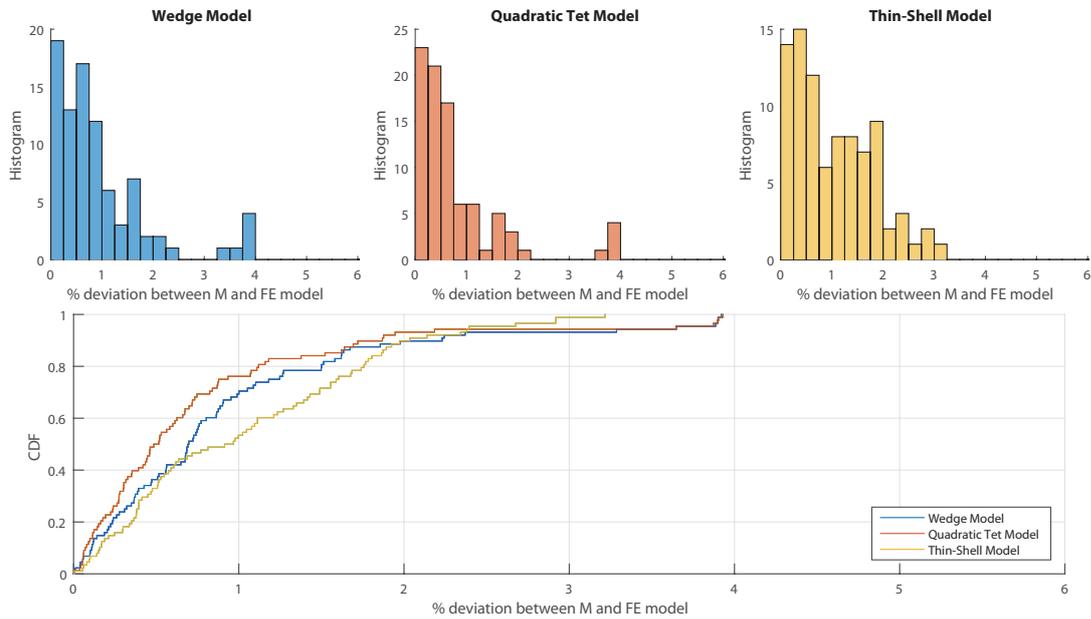


Figure 5.5: Histogram and empirical CDF of the relative deviations between the measured frequencies and the finite element model frequencies with the optimized material parameters.

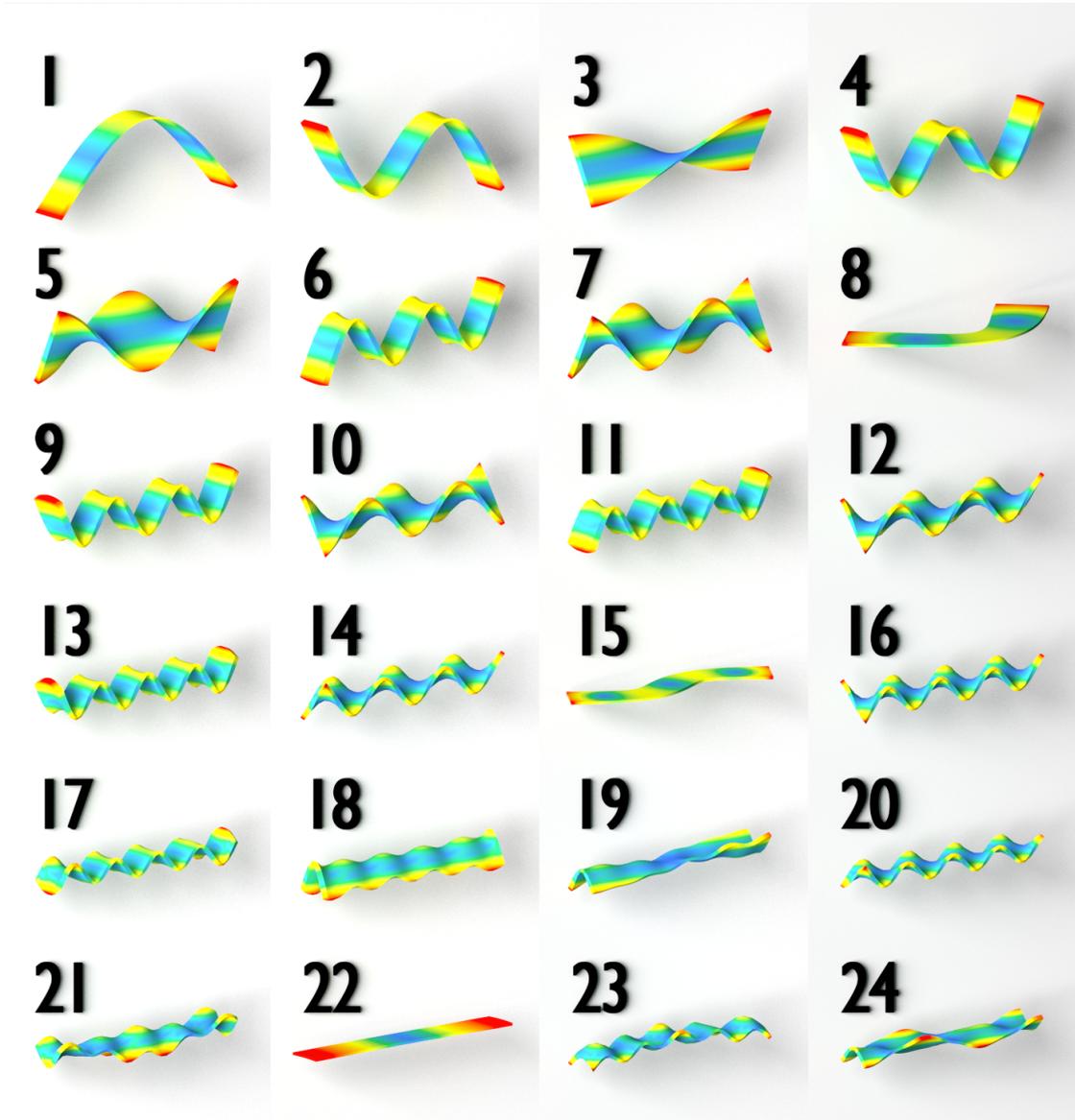
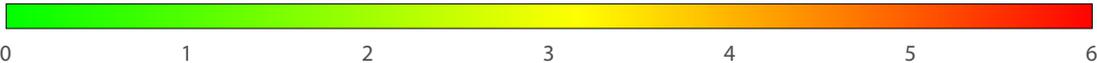


Figure 5.6: The first 24 mode shapes of aluminium plate 3.

Mode No.	Plate 0					Plate 1				
	M	W	QT	TS	LT	M	W	QT	TS	LT
1	2987	3008	3012	3021	8296	1152	1176	1176	1178	3279
2	-	4481	4499	4580	13120	-	2782	2792	2841	8021
3	8260	8332	8361	8351	21723	3191	3257	3262	3287	8885
4	-	9627	9686	9971	x	-	5773	5800	5919	13791
5	15820	16087	16211	16416	x	6271	6407	6428	6461	16422
6	15977	16144	16290	16660	x	9007	9159	9216	9390	17020
7	19472	19499	19612	19741	x	10362	10600	10655	10481	x
8	20798	20959	21110	21398	x	12850	13107	13211	13504	x
9	x	x	x	x	x	13515	13301	13300	13801	x
10	x	x	x	x	x	15400	15762	15880	16053	x
11	x	x	x	x	x	17358	17757	17937	18351	x
12	x	x	x	x	x	19344	19572	19668	19653	x
13	x	x	x	x	x	19676	19952	20063	20096	x
14	x	x	x	x	x	x	x	x	21255	x
15	x	x	x	x	x	x	x	x	x	x
16	x	x	x	x	x	x	x	x	x	x
17	x	x	x	x	x	x	x	x	x	x
18	x	x	x	x	x	x	x	x	x	x
19	x	x	x	x	x	x	x	x	x	x
20	x	x	x	x	x	x	x	x	x	x
21	x	x	x	x	x	x	x	x	x	x
22	x	x	x	x	x	x	x	x	x	x
23	x	x	x	x	x	x	x	x	x	x
24	x	x	x	x	x	x	x	x	x	x

Table 5.3: This table displays the natural frequencies determined by the different finite element models for the rectangular aluminium plates with IDs 0 and 1 using listed parameter values without parameter estimation. All frequencies are in Hz. Frequencies outside the range 20–20,000 Hz are marked with an ‘x’. M = Measured frequencies, W = Wedge elements, QT = Quadratic tetrahedral elements, TS = Thin-shell elements, LT = Linear tetrahedral elements.

Mode No.	Plate 2					Plate 3				
	M	W	QT	TS	LT	M	W	QT	TS	LT
1	624	636	636	637	1810	517	523	523	524	1487
2	1725	1760	1761	1766	4943	1426	1447	1448	1453	4069
3	-	2027	2034	2069	5917	-	1836	1843	1875	5351
4	3391	3463	3469	3482	8063	2805	2847	2851	2860	6749
5	4088	4148	4165	4238	9573	3697	3745	3760	3823	7900
6	5621	5741	5760	5834	11997	4651	4721	4733	4796	10818
7	6350	6449	6482	6595	15582	5715	5793	5820	5917	12905
8	7876	7750	7749	8036	18403	6590	6483	6483	6720	16295
9	8401	8591	8631	8693	19102	6955	7067	7095	7146	16531
10	8855	9005	9062	9224	x	7920	8038	8085	8223	18988
11	11719	11883	11974	12189	x	9714	9879	9933	10005	20570
12	-	11997	12074	12189	x	10360	10533	10608	10775	x
13	14799	15139	15281	15555	x	12918	13144	13235	13375	x
14	15560	15918	16049	16241	x	13082	13324	13435	13688	x
15	18363	18450	18447	18980	x	15980	15721	15723	15687	x
16	19005	18818	19032	19394	x	16133	16447	16616	16888	x
17	19220	19379	19472	19614	x	16537	16806	16951	18371	x
18	-	19386	19517	19705	x	19241	19256	19371	19575	x
19	20400	20820	21022	20840	x	19705	19722	19835	20286	x
20	20780	21038	21177	21495	x	-	19932	20174	20286	x
21	x	x	x	x	x	20305	20452	20583	20600	x
22	x	x	x	x	x	-	20570	20595	21104	x
23	x	x	x	x	x	21292	21526	21726	x	x
24	x	x	x	x	x	x	x	x	x	x



0 1 2 3 4 5 6

% deviation between M and FE model

Table 5.4: This table displays the natural frequencies determined by the different finite element models for the rectangular aluminium plates with IDs 2 and 3 using listed parameter values without parameter estimation. All frequencies are in Hz. Frequencies outside the range 20–20,000 Hz are marked with an ‘x’. M = Measured frequencies, W = Wedge elements, QT = Quadratic tetrahedral elements, TS = Thin-shell elements, LT = Linear tetrahedral elements.

Mode No.	Plate 4					Plate 5				
	M	W	QT	TS	LT	M	W	QT	TS	LT
1	335	340	340	341	969	232	236	236	236	674
2	926	941	941	941	2657	640	652	652	654	1851
3	1470	1482	1487	1514	4306	-	1233	1237	1258	3190
4	1820	1850	1852	1858	4519	1253	1280	1282	1284	3546
5	-	3004	3016	3075	5182	2078	2122	2125	2128	3661
6	3017	3067	3073	3075	8389	-	2489	2498	2541	5923
7	4393	4334	4334	4490	8794	-	3057	3057	3167	7242
8	4520	4594	4606	4631	11309	3106	3179	3185	3195	8182
9	-	4604	4624	4704	12444	3723	3792	3807	3873	8783
10	6319	6316	6348	6457	13327	4351	4450	4462	4488	10973
11	-	6429	6453	6513	16604	5062	5164	5188	5275	12158
12	8053	8171	8219	8363	17127	5805	5935	5956	5987	13830
13	8417	8570	8611	8667	18142	6487	6623	6657	6770	14712
14	10031	10194	10265	10471	19876	7459	7632	7666	7626	14835
15	10808	10886	10884	10898	x	-	7861	7860	8086	16019
16	-	11011	11077	11336	x	8027	8187	8235	8375	18865
17	12190	12411	12510	12736	x	9317	9541	9592	9659	20333
18	13487	13743	13845	13951	x	9642	9871	9937	10106	x
19	14552	14840	14977	15247	x	11371	11656	11734	11451	x
20	-	16605	16606	15922	x	-	11688	11780	11955	x
21	-	16737	16882	16658	x	13306	13653	13771	13878	x
22	17130	17501	17685	17959	x	13642	13829	13831	14007	x
23	-	19188	19189	19765	x	-	13973	14082	14223	x
24	-	19283	19393	19765	x	14080	14174	14167	14696	x

0 1 2 3 4 5 6
% deviation between M and FE model

Table 5.5: This table displays the natural frequencies determined by the different finite element models for the rectangular aluminium plates with IDs 4 and 5 using listed parameter values without parameter estimation. All frequencies are in Hz. Frequencies outside the range 20–20,000 Hz are marked with an ‘x’. M = Measured frequencies, W = Wedge elements, QT = Quadratic tetrahedral elements, TS = Thin-shell elements, LT = Linear tetrahedral elements.

Mode No.	Plate 0					Plate 1				
	M	W	QT	TS	LT	M	W	QT	TS	LT
1	2987	2939	2942	2950	8103	1152	1148	1149	1150	3203
2	-	4377	4395	4474	12815	-	2718	2727	2776	7834
3	8260	8136	8168	8335	21218	3191	3181	3186	3194	8679
4	-	9400	9462	9655	x	-	5639	5665	5765	13470
5	15820	15712	15836	15973	x	6271	6258	6278	6316	16040
6	15977	15775	15910	16226	x	9007	8946	9001	9165	16624
7	19472	19087	19156	19351	x	10362	10351	10408	10506	x
8	20798	20469	20615	20912	x	12850	12803	12906	13158	x
9	x	x	x	x	x	13515	12990	12992	13451	x
10	x	x	x	x	x	15400	15394	15510	15696	x
11	x	x	x	x	x	17358	17341	17520	17864	x
12	x	x	x	x	x	19344	19098	19216	19455	x
13	x	x	x	x	x	19676	19459	19599	20024	x
14	x	x	x	x	x	-	21904	x	x	x
15	x	x	x	x	x	x	x	x	x	x
16	x	x	x	x	x	x	x	x	x	x
17	x	x	x	x	x	x	x	x	x	x
18	x	x	x	x	x	x	x	x	x	x
19	x	x	x	x	x	x	x	x	x	x
20	x	x	x	x	x	x	x	x	x	x
21	x	x	x	x	x	x	x	x	x	x
22	x	x	x	x	x	x	x	x	x	x
23	x	x	x	x	x	x	x	x	x	x
24	x	x	x	x	x	x	x	x	x	x

0 1 2 3 4 5 6
% deviation between M and FE model

Table 5.6: This table displays the natural frequencies determined by the different finite element models for the rectangular aluminium plates with IDs 0 and 1 using estimated parameter values. All frequencies are in Hz. Frequencies outside the range 20–20,000 Hz are marked with an ‘x’. M = Measured frequencies, W = Wedge elements, QT = Quadratic tetrahedral elements, TS = Thin-shell elements, LT = Linear tetrahedral elements.

Mode No.	Plate 2					Plate 3				
	M	W	QT	TS	LT	M	W	QT	TS	LT
1	624	621	621	622	1768	517	511	511	511	1453
2	1725	1719	1720	1717	4828	1426	1413	1414	1418	3974
3	-	1980	1987	2021	5780	-	1794	1800	1831	5226
4	3391	3382	3389	3401	7875	2805	2780	2785	2794	6592
5	4088	4052	4068	4140	9350	3697	3658	3672	3737	7716
6	5621	5608	5626	5721	11718	4651	4611	4623	4643	10566
7	6350	6299	6331	6440	15219	5715	5658	5685	5784	12605
8	7876	7569	7569	7848	17975	6590	6332	6331	6563	15916
9	8401	8392	8431	8495	18658	6955	6902	6931	6976	16147
10	8855	8795	8851	9009	x	7920	7852	7896	8038	18546
11	11719	11607	11696	11894	x	9714	9647	9702	9705	20091
12	-	11722	11794	12073	x	10360	10288	10361	10535	x
13	14799	14790	14928	15231	x	12918	12834	12925	13053	x
14	15560	15544	15676	15877	x	13082	13014	13122	13362	x
15	18363	18022	18020	18294	x	15980	15357	15356	16209	x
16	19005	18380	18590	18941	x	16133	16064	16224	16209	x
17	19220	18932	19014	19256	x	16537	16412	16566	16511	x
18	-	18943	19063	19587	x	19241	18809	18910	19143	x
19	20400	20329	20527	21056	x	19705	19265	19371	19545	x
20	20780	20554	20684	21056	x	-	19465	19683	20083	x
21	x	x	x	x	x	20305	19976	20087	20306	x
22	x	x	x	x	x	-	20093	20114	20306	x
23	x	x	x	x	x	21292	21026	21222	21510	x
24	x	x	x	x	x	-	21951	x	x	x

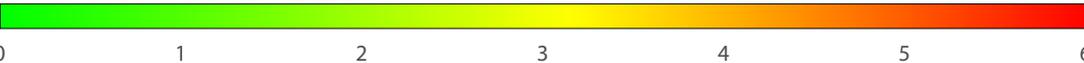


0 1 2 3 4 5 6

% deviation between M and FE model

Table 5.7: This table displays the natural frequencies determined by the different finite element models for the rectangular aluminium plates with IDs 2 and 3 using estimated parameter values. All frequencies are in Hz. Frequencies outside the range 20–20,000 Hz are marked with an ‘x’. M = Measured frequencies, W = Wedge elements, QT = Quadratic tetrahedral elements, TS = Thin-shell elements, LT = Linear tetrahedral elements.

Mode No.	Plate 4					Plate 5				
	M	W	QT	TS	LT	M	W	QT	TS	LT
1	335	332	333	333	946	232	230	230	231	658
2	926	919	919	921	2596	640	636	637	637	1808
3	1470	1448	1453	1477	4206	-	1204	1208	1229	3116
4	1820	1807	1809	1813	4414	1253	1251	1252	1254	3463
5	-	2935	2945	3000	5062	2078	2073	2077	2074	3575
6	3017	2996	3001	3014	8194	-	2431	2439	2481	5786
7	4393	4233	4233	4386	8589	-	2986	2984	3093	7074
8	4520	4487	4499	4518	11046	3106	3105	3111	3122	7992
9	-	4497	4517	4593	12154	3723	3704	3722	3783	8579
10	6319	6169	6200	6303	13017	4351	4346	4361	4334	10718
11	-	6279	6303	6368	16218	5062	5043	5066	5153	11875
12	8053	7980	8028	8181	16728	5805	5797	5832	5850	13508
13	8417	8370	8410	8468	17720	6487	6469	6498	6615	14370
14	10031	9957	10025	10219	19413	7459	7455	7480	7532	14490
15	10808	10631	10631	10916	21807	-	7679	7656	7965	15646
16	-	10754	10820	11025	x	8027	7996	8008	8179	18426
17	12190	12122	12219	12417	x	9317	9318	9371	9443	19860
18	13487	13423	13523	13637	x	9642	9641	9669	9871	21641
19	14552	14495	14629	14941	x	11371	11385	11470	11354	x
20	-	16220	16219	16237	x	-	11416	11522	11711	x
21	-	16347	16490	16574	x	13306	13336	13489	13513	x
22	17130	17094	17274	17531	x	13642	13507	13489	13694	x
23	-	18742	18743	19288	x	-	13650	13721	13903	x
24	-	18836	18948	19288	x	14080	13844	13806	14316	x



0 1 2 3 4 5 6

% deviation between M and FE model

Table 5.8: This table displays the natural frequencies determined by the different finite element models for the rectangular aluminium plates with IDs 4 and 5 using estimated parameter values. All frequencies are in Hz. Frequencies outside the range 20–20,000 Hz are marked with an ‘x’. M = Measured frequencies, W = Wedge elements, QT = Quadratic tetrahedral elements, TS = Thin-shell elements, LT = Linear tetrahedral elements.

5.2 Results

In order to optimize and produce an instrument, it is important to choose a suitable material and fabrication method. In the family of modern struck idiophones, we encounter a large variety of different materials: Wood, especially rosewood and padauk, has traditionally been used to produce marimbas and xylophones. A modern trend is to use synthetic fiberglass or fiberglass-reinforced plastics instead. A different class of idiophones is that of metallophones. Common materials are steel, which is used for glockenspiels triangles, and steel drums; brass, which is used for tubular bells; aluminium, for the bell lyra; and bronze, for instruments such as the Gamelan, cymbals, tam-tams, and bells [FR98].

The difficulty in making metallic instruments is that the material must be able to be cast into a complex shape and, at the same time, be hard enough to emit pleasant sounds. Additional criteria are durability, attractive appearance, availability, and inexpensiveness. This is why the choice of materials for instruments was very limited up until the eighteenth century. Instruments older than that were usually produced from native metals such as gold, silver, and meteoritic iron. Other materials include lead, tin, zinc, and copper.

Our first experiment in fabricating an instrument was to 3d-print circular plates from PLA, polylactic acid, the plastic used by our FDM printer. However, recording sound samples and analyzing the spectrogram revealed that the material is too soft and damping to exhibit an intelligible pitch. Another difficulty with this material is the microstructure introduced by the FDM printing process: Young's modulus has a dependence on the direction of the material paths, which makes it difficult to apply an isotropic constitutive model.

An important factor for our choice of material is quick prototyping, i.e., short production cycles from an optimized 3d model to the finished physical artifact. Common metals such as brass, steel, and aluminium are difficult to process for hobbyists because they have melting points of 600°C and above. This leaves us with tin and lead, which melt at 230°C and 330°C respectively. Out of those two, tin is a better choice for us because its Young's modulus of 50 GPa is significantly higher than the Young's modulus of lead, which is 16 GPa. This makes for a harder material with better sound characteristics.

In the remainder of this chapter, we describe the production process of two differently shaped bells and compare the measured pitch to the computational solution. The first bell consists of a hemiellipsoidal shell and is therefore similar to the design of temple bells. This smooth geometry enables a clear ringing sound with low damping and is easy to analyze because of its low curvature. Therefore, it is a good candidate to test per-vertex-normal parametrization. In the second experiment, we determine how well our pipeline can work with a more complex shape. We design a bell in the shape of a thin-shelled rabbit that has regions of high curvature as well as a geometrically smooth region that acts as a resonator. Due to the high curvatures, per-vertex-normal parametrization is not applicable, and we use skeleton parametrization instead.

Additionally, we describe an application of manifold harmonics parametrization to the optimization of harmonic ratios in a marimba bar. This result is purely computational,

and no physical prototype has been fabricated.

5.2.1 Hemiellipsoid Bell

In the manufacturing process, we liquify the tin in a small melting furnace and pour it into a mold to give it a complex shape. The mold must be able to resist the temperature of the molten tin, and therefore it cannot be 3d-printed from PLA directly. For simple shapes, we use oiled molding sand to create the molds. This requires a positive of the 3d object, which we can produce with an FDM printer.

The whole production process is documented in Figs. 5.7 and 5.8. To start with, the positive is placed in a plastic frame with the circular edge down, brushed with separation powder to keep the sand from sticking, and covered with molding sand, layer by layer. Each layer is compressed manually with a wooden block before proceeding with the next layer. Once the frame has been filled up with sand, the upper half of the sand mold is finished. The block of sand, including the positive, is then turned over, and the bottom surface is brushed with separation powder to keep the two sand blocks from sticking together. A second plastic frame is fixed to the first by inserting metal rods into the dedicated holes, and the process of layering and compressing sand is repeated. The two halves can then be separated, and the model is removed. The upper half of the negative is equipped with a feed opening in the middle for the molten tin, and small holes through which the air can escape. Finally, the two halves are joined back together to form the finished mold. Alignment is ensured by reinserting the metal rods that keep the two plastic frames together.

Tin is melted in a melting furnace and poured into the feed opening. As soon as the tin rises through the feed opening and the airholes, the mold is completely filled. After a cooling period of ten minutes, the sand mold can be destroyed to recover the tin model. The appendages resulting from the feed opening and the airholes have to be sawed off. Any inaccuracies can be corrected by filing off bits of the model.

For the production of our first bell, we chose a hemiellipsoidal shape. For this smooth, low-curvature shape, it suffices to use the naive per-vertex-normal parametrization with a single optimization parameter, see Section 4.2.2, as illustrated in Fig. 5.9. Although very large offsets can lead to self-intersections, it is easy to set a global upper bound on the offset magnitudes to avoid cases like in the rightmost figure.

Parameter	Tin #1
Density ρ [kg/m ³]	7280
Young's Modulus E [GPa]	49.9
Poisson's ratio ν	0.36

Table 5.9: Material properties for the tin used for the hemiellipsoidal bell.

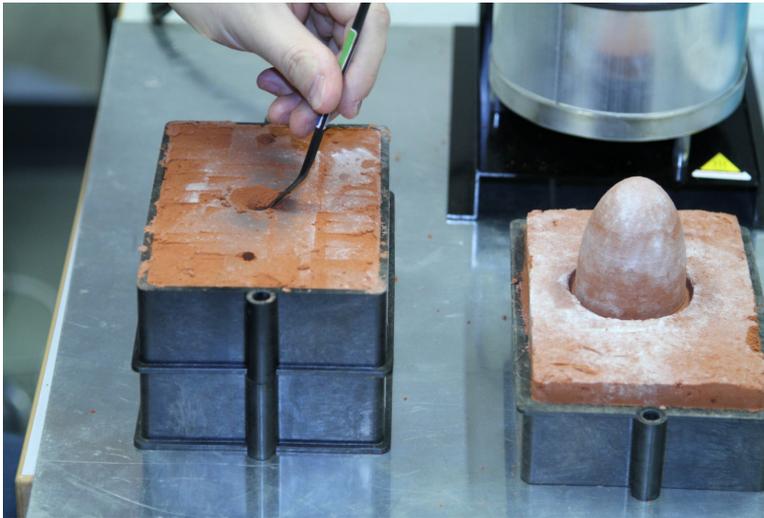
The outer surface of the bell has a height of 6 cm, a base width of about 5.3 cm, and the mesh consists of 341 vertices and 640 faces. The raw tin bars we use for production



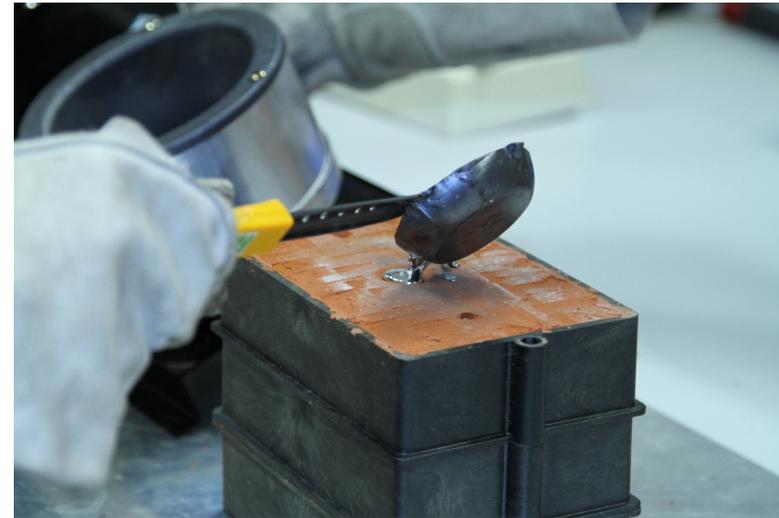
(a) After the upper half is finished, we begin to make the lower half.



(b) A layer of the lower half is being compressed.



(c) The upper half is equipped with a feed opening.



(d) Molten tin is poured into the mold.

Figure 5.7: Production of a tin model with oiled molding sand.



(a) The mold is destroyed to recover the object.



(b) The tin appendages are sawed off.



(c) The positive and the finished model side by side.



(d) The suspended bell, ready to record sound samples.

Figure 5.8: Finishing the tin bell and recording sound samples.

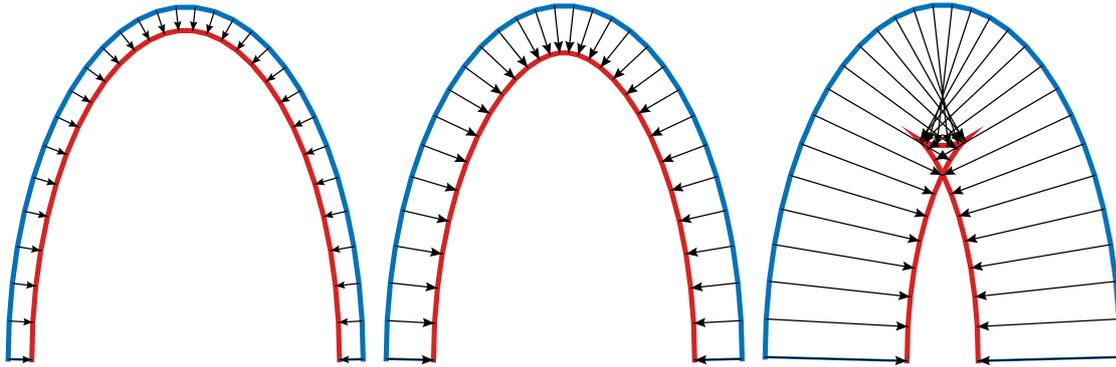


Figure 5.9: Offset surfaces of a hemiellipsoid using per-vertex normals, cross-sectional view. Offset magnitudes increase from left to right. The right image illustrates that self-intersections can occur if the offset magnitude exceeds the curvature radius.

are of Sn99.9 purity grade, and the material parameters obtained from a data sheet are listed in Table 5.9.

For this bell, we use the thin-shell finite element model for the calculation of the pitch for two reasons. Firstly, thin-shell elements are well suited to this shape because the angles between adjacent elements are small, making the shape approximately flat locally. Secondly, we experience shorter run-times using the thin-shell model due to the small size of the finite element matrices and the faster solution to the eigenproblem. As a preliminary, we perform a sweep of the parameter space and find out that a wall thickness of 2 mm yields a pitch of 1104 Hz (approximately a D \flat 6) and a wall thickness of 6 mm yields a pitch of 3270 Hz (approximately an A \flat 7). For our bell we aim for a pitch of 1760 Hz (A6). Using 2 mm as the initial wall thickness parameter value and setting `DiffMinChange` to $2 \cdot 10^{-5}$, the non-linear optimization routine finds an optimal solution of 3.2 mm after 25 seconds.

We manufacture a physical copy of the bell as described above and experimentally determine the frequency spectrum. Table 5.10 shows the measured frequencies and the corresponding frequencies obtained with the thin-shell finite element model. Note that the solution predicted by the thin-shell model is 1761 instead of 1760 due to numerical issues in the evaluation of the objective function, as discussed in Section 4.4. Additionally, the table shows the finite element results using wedge and quadratic tetrahedral elements for the same bell geometry. The reason we show the results for these models as well is because it allows us to estimate how much of the error between the measured and the computed frequencies is due to the production method, and how much is due to the error of the finite element model. Note that the frequencies appear with a multiplicity of two because the triangle mesh used to represent the bell is almost, but not entirely, symmetric.

The measured pitch of the bell is 1809 Hz, which has a relative error of 2.8% w.r.t. the desired pitch of 1760 Hz. The error is larger than what was achieved for the aluminium plates, see Section 5.1. It is impossible to separate the error of the finite element model

Mode No.	Hemiellipsoid Bell			
	M	TS	W	QT
1	1809	1761	1736	1744
2	1809	1762	1736	1744
3	4950	4804	4689	4723
4	4950	4830	4689	4723
5	8666	9031	8672	8754
6	8666	9164	8675	8760
7	9097	9283	9161	9169
8	9097	9295	9164	9173
9	10460	10899	10573	10639
10	10460	10899	10594	10641
11	12112	12462	12503	12488
12	12112	12462	12552	12507

Table 5.10: This table displays the first twelve natural frequencies determined by the different finite element models for the hemiellipsoidal bell. All frequencies are in Hz. M = Measured frequencies, TS = Thin-shell elements, W = wedge elements, QT = quadratic tetrahedral elements.

and the error due to the production process in a definite way. However, we observe that the deviation among the pitches as predicted by the different finite element models is less than 1.5%. Therefore, it is reasonable to assume that no more than half of the total error of 2.8% is due to the finite element approximation. If the actual wall thickness of the fabricated bell deviates by as little as 0.1 mm from the desired wall thickness of 3.2 mm, the resulting pitch will deviate from the desired pitch by more than 3%. It is likely that an error of this magnitude was introduced when creating the sand molds. Therefore, the error of 2.8% can also be attributed to the fabrication process. Additionally, the available material parameters for Sn99.9 may deviate from the real parameters. For the batch of tin that we used, we cannot perform parameter optimization, because the whole batch was used up in the production of the bell.

5.2.2 Rabbit Bell

In this section we explore the fabrication of a bell with a shape that is more complex than a hemiellipsoid. We use the rabbit mesh shown in Fig. 5.10a as the outer surface. The mesh has 4,300 vertices and 8,482 faces and is thus more finely resolved than the mesh for the hemiellipsoid bell.

For this object, the naive offsetting approach along the per-vertex normals does not work anymore because it produces overfolds even for small offset magnitudes, as illustrated in Fig. 5.11 with a 2d example. To counter this behavior, we use the skeleton

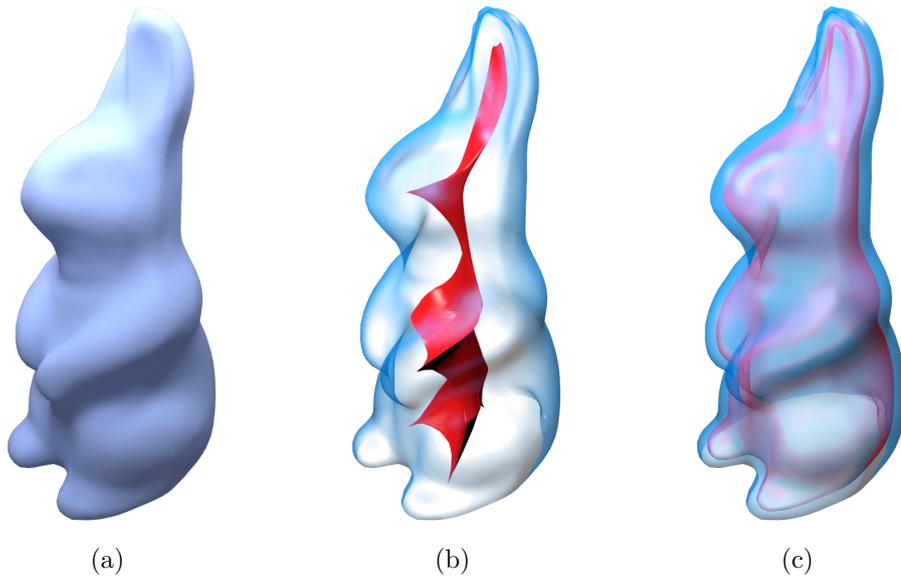


Figure 5.10: The triangle mesh used for the rabbit bell. (a) Outer surface. (b) Transparent outer surface and meso-skeleton. (c) Outer surface and optimized inner surface.

parametrization with a single optimization parameter as introduced in Section 4.2.3. In a preprocessing step, the rabbit mesh is contracted to form a meso-skeleton, i.e., a structure consisting of 1d path segments and 2d surface sheets, see Fig. 5.10b. The right image in Fig. 5.12 illustrates the effect of using the skeleton on large offset magnitudes. The offset magnitude at vertices in high-curvature regions is clamped so as to not go beyond the skeleton.

Choosing a thickness parameter of 2 mm yields a pitch frequency of 2210 Hz (approximately D \flat 7), and 6 mm yields 5450 Hz (approximately G8). We optimize the rabbit bell for the tone E7, which equals a pitch of 2637 Hz. After eight minutes, the non-linear optimization routine finds an optimal thickness parameter of 2.48 mm using the wedge finite element model. We do not use the thin-shell model for this mesh because the high-curvature regions around the ears of the rabbit are better captured using continuous rather than structural elements. Additionally, the evaluation of the wedge model yields shorter run-times due to the faster construction of the finite element matrices. The corresponding outer and inner surfaces are illustrated in Fig. 5.10c.

Producing the rabbit bell with sand molds is not possible because both the outer and the inner surface cannot be represented as a height field. This means that the plastic rabbit model could not be removed from the molds after the sand has been compressed because it would rip out chunks of sand. We opt to use silicon caoutchouc, or natural rubber, instead because it is flexible enough to remove from the plastic shell. The manufacturing process is illustrated in Figs. 5.13 and 5.14. We design a three-part caoutchouc mold for the rabbit model, one inner mold and a two-part outer mold. For each of the outer mold parts, we design a single plastic mold. For the inner caoutchouc

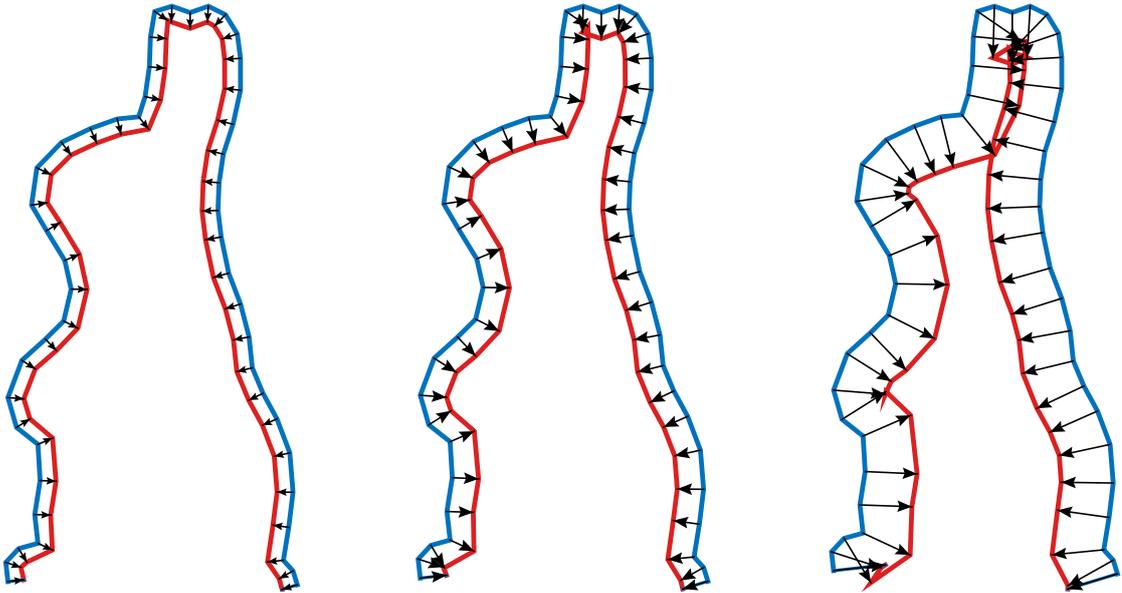


Figure 5.11: Offset surfaces of the rabbit mesh using per-vertex normals, cross-sectional view. Offset magnitudes increase from left to right. The center image illustrates that even for moderate offset magnitudes, self-intersections occur in high-curvature regions like the ears and the feet.

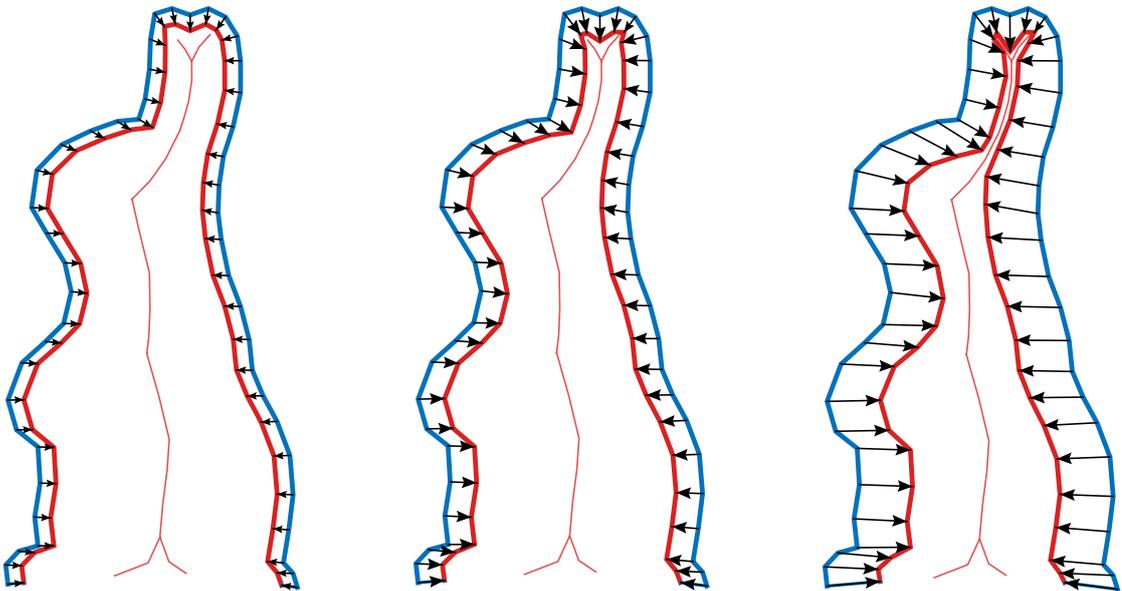


Figure 5.12: Offset surfaces of a rabbit mesh using the skeleton parameterization, cross-sectional view. Offset magnitudes increase from left to right. The right image illustrates that self-intersections are avoided even for high offset magnitudes. This is achieved by limiting the offset vectors to never cross the skeleton.

mold, we use a two-part plastic mold. All four plastic molds are produced from PLA with an FDM printer.

The plastic molds are first brushed with petroleum jelly to keep the caoutchouc from sticking. The liquid caoutchouc is then mixed with a cross-linking agent. This agent links the polymer chains in the caoutchouc and completely solidifies it after several hours. We brush the plastic molds with a layer of caoutchouc before filling them completely in order to avoid air bubbles. After the caoutchouc molds have solidified, they can be removed from their plastic shells and assembled.

Once again, tin is melted in the melting furnace and poured into the mold. After ten minutes, we remove the outer parts of the mold. The inner part of mold has to be destroyed in order to remove it because it is too stiff to pull out in one piece.

A new batch of tin from a different producer was used in the production of the rabbit bell. There is no reliable data sheet available to us, but this batch of tin is also labeled “pure tin” like the one used for the hemiellipsoid bell. However, this second batch feels significantly softer and produces the typical crackling sound known as the tin cry when deformed. To optimize the rabbit bell, we use the same material parameters as for the hemiellipsoid bell, see the “Listed” column in Table 5.11.

Parameter	Tin #2	
	Listed	Estimated
Density ρ [kg/m ³]	7280	7419
Young’s Modulus E [GPa]	49.9	46.2
Poisson’s ratio ν	0.36	0.36

Table 5.11: Material properties for tin used for the rabbit bell.

The measured frequencies are listed in Table 5.12, and this time we observe a significantly higher error in the prediction of the pitch. The experimentally determined pitch has a frequency of 2369 Hz (approximately the note D7), which deviates from the desired E7 by a whole tone, or 11%. In order to explain this large deviation, we produce two rectangular thin plates out of tin for material parameter estimation. The thin plates are produced by making sand molds from the aluminium plates 1 and 2 used in Section 5.1.3, and casting them in tin.

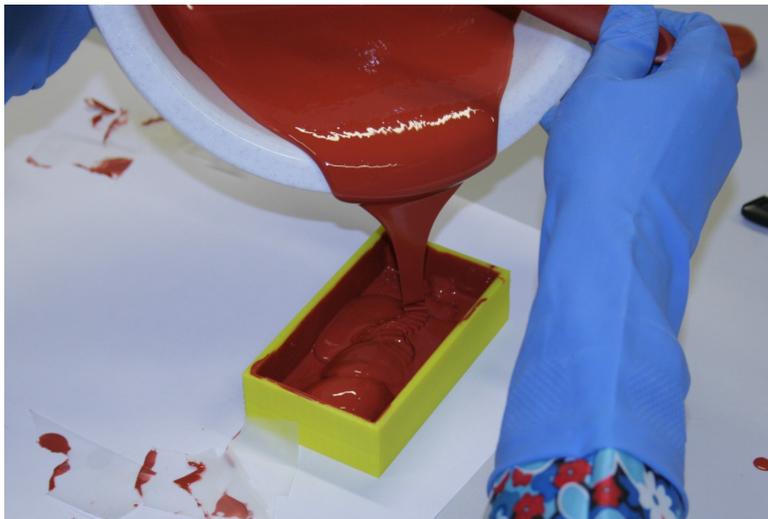
The tin plates have visible manufacturing flaws where the feed opening and airholes appendages have been sawed and rasped off. Therefore we cannot expect results as good as for the aluminium plates. The experimentally determined frequencies and the finite element prediction for the listed material parameters are documented in Table 5.13. Note that the color key reflects a larger range in frequency deviation, namely green for 0%, yellow for 6%, and red for 12%. The material parameters after running the estimation procedure are listed in the “Estimated” column in Table 5.11. Evaluating the finite elements model again with the estimated parameters yields the frequencies listed in Table 5.14. It is evident that the frequencies for the short tin plate are now captured adequately, while the frequencies for the long tin plate are still far off.



(a) The plastic molds for the caoutchouc molds.



(b) Brushing plastic molds with caoutchouc.

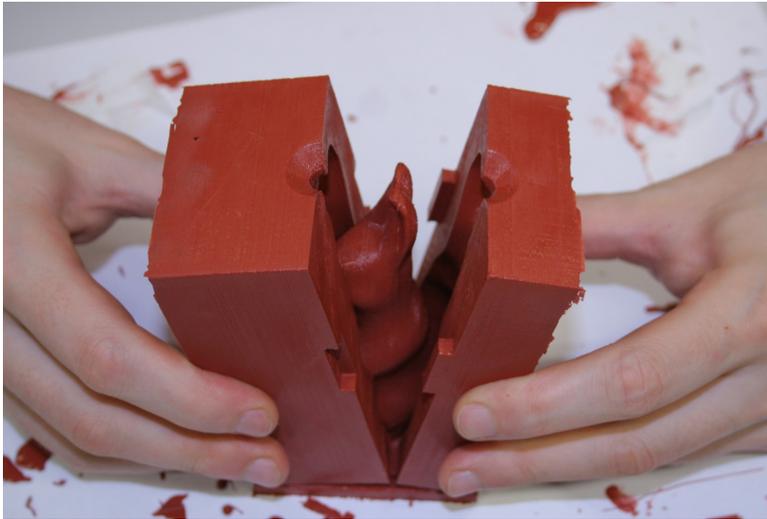


(c) Pouring caoutchouc into the molds.



(d) One part of the caoutchouc mold.

Figure 5.13: Production of the rabbit bell with caoutchouc molds.



(a) Closing the three-part caoutchouc mold.



(b) The tin rabbit after removing the outer molds.



(c) Removing the inner caoutchouc mold.



(d) The finished rabbit model.

Figure 5.14: Finishing the rabbit bell.

Mode No.	Listed Parameters				Estimated Parameters			
	M	W	QT	TS	M	W	QT	TS
1	2369	2637	2647	2706	2369	2513	2522	2573
2	2491	2756	2770	2830	2491	2628	2639	2699
3	2797	4323	4338	4238	2797	4120	4134	4026
4	3545	4941	4956	4850	3545	4708	4722	4618
5	4742	5256	5277	5242	4742	5010	5028	5024
6	4848	5403	5429	5455	4848	5149	5173	5209
7	4983	6062	6102	6523	4983	5782	5816	6063
8	5409	6414	6447	6790	5409	6113	6143	6442
9	6360	7152	7187	7002	6360	6812	6849	6921
10	6663	7304	7318	7519	6663	6955	6972	6964

Table 5.12: This table displays the natural frequencies determined by the different finite element models for the rabbit bell. The evaluation as been performed with the listed parameter values and with the estimated material parameters. All frequencies are in Hz. M = Measured frequencies, W = Wedge elements, QT = Quadratic tetrahedral elements, TS = Thin-shell elements.

The right part of Table 5.12 shows the predicted frequencies for the tin rabbit using the estimated material parameters. This improves the error in pitch prediction of the wedge model from 11% to 6%, which is approximately a semitone.

5.2.3 Harmonic Ratio Optimization

In every high-quality instrument, a resonator is not only designed to match a particular pitch but also to exhibit a harmonic ratio in its overtones. Depending on the instrument, different harmonic ratios are being used. For example, the torsional modes of a glockenspiel have the frequency ratios 1:2:3. Carillons, i.e., church bells, are built such that the first and second mode are one octave apart, the second and third mode are a minor third apart, the second and fourth mode are a fifth apart, and the second and fifth mode are an octave apart.

Marimba bars are equipped with an arched carving on the bottom side. This design serves two functions. Firstly, it increases all frequencies in the mode spectrum, and thus reduces the length of bar necessary to reach low notes. Secondly, it allows tuning the overtones of the bar, most importantly a ratio of 1:4 between the first and the second mode [FR98].

This section demonstrates how the manifold harmonics parametrization can be used to optimize the shape of a marimba bar w.r.t. two simultaneous goals, namely, to match the pitch to the desired frequency and to create a 1:4 ratio between the pitch and the first overtone. As opposed to the two previous sections, the result presented here is purely

Mode No.	Short Tin Plate				Long Tin Plate			
	M	W	QT	TS	M	W	QT	TS
1	610	631	632	633	288	337	337	337
2	1461	1462	1467	1484	831	932	934	930
3	1645	1751	1754	1806	-	1062	1066	1085
4	2790	3038	3053	3125	1626	1837	1841	1847
5	3201	3448	3462	3490	2035	2175	2185	2232
6	4492	4833	4866	4898	2682	3048	3059	3038
7	5320	5707	5744	5800	3106	3386	3404	3449
8	6568	6937	6998	7365	3684	4070	4070	4223
9	6868	7090	7092	7486	4005	4563	4588	4637
10	8020	8478	8558	8532	4293	4733	4766	4860
11	8884	9426	9536	9702	5567	6254	6307	6425
12	9655	10396	10461	10335	5719	6374	6422	6540

0 2 4 6 8 10 12
% deviation between M and FE model

Table 5.13: This table displays the natural frequencies determined by the different finite element models for the rectangular tin plates with the listed material parameters. All frequencies are in Hz. M = Measured frequencies, W = Wedge elements, QT = Quadratic tetrahedral elements, TS = Thin-shell elements.

Mode No.	Short Tin Plate				Long Tin Plate			
	M	W	QT	TS	M	W	QT	TS
1	610	601	602	603	288	321	321	321
2	1461	1393	1398	1423	831	889	890	894
3	1645	1668	1672	1670	-	1012	1016	1033
4	2790	2895	2909	2964	1626	1750	1754	1762
5	3201	3285	3298	3320	2035	2073	2081	2118
6	4492	4605	4636	4730	2682	2904	2915	2914
7	5320	5438	5474	5451	3106	3226	3243	3279
8	6568	6610	6669	6817	3684	3878	3878	4029
9	6868	6756	6756	7012	4005	4348	4372	4403
10	8020	8078	8153	8270	4293	4510	4541	4630
11	8884	8981	9089	9298	5567	5958	6009	6104
12	9655	9906	9966	10090	5719	6074	6120	6230

0 2 4 6 8 10 12
% deviation between M and FE model

Table 5.14: This table displays the natural frequencies determined by the different finite element models for the rectangular tin plates with the estimated material parameters. All frequencies are in Hz. M = Measured frequencies, W = Wedge elements, QT = Quadratic tetrahedral elements, TS = Thin-shell elements.

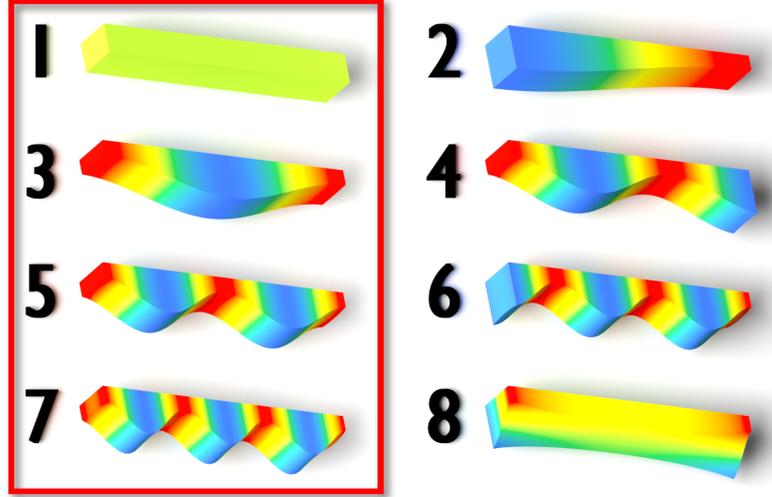


Figure 5.15: The influence regions of the first eight eigenfunctions of the Laplacian of the marimba bar surface. Only the eigenfunctions 1, 3, 5, and 7 are symmetric about the bisecting plane.

Parameter	East Indian Rosewood
Density ρ [kg/m ³]	900
Young's Modulus E [GPa]	11.5
Poisson's ratio ν	0.33

Table 5.15: Material properties for East Indian Rosewood.

computational, and no physical marimba bar has been produced. We use the wedge model and the same finite element mesh as for the aluminium plates, i.e., a single layer of wedge elements obtained by triangulating a regular grid with a cell width of 1 cm. We aim for a pitch of 523.3 Hz, which is a high C. The marimba bar is represented as the volume enclosed by a top surface and a bottom surface. The length and width of the bar are controlled via two optimization parameters w and l . The shape of the top surface is fixed to be a flat rectangle, but the shape of the bottom surface is allowed to change by varying the coefficients α of the Laplacian eigenfunctions.

We use a total of four eigenfunctions to modulate the shape of the undercut. Because we aim for a symmetric undercut, we pick the four lowest-frequency symmetric eigenfunctions and omit asymmetric ones. Fig. 5.15 illustrates the influence of the first eight eigenfunctions on the per-vertex-displacement magnitudes, with the symmetric eigenfunctions highlighted in a red frame. The first eigenfunction is constant and its coefficient directly controls the thickness of the bar. The coefficients of eigenfunctions 3, 5, and 7 control the shape of the bottom surface. This yields a six-dimensional design space parametrized by $(w, l, \alpha_1, \alpha_3, \alpha_5, \alpha_7)$, where we denote $\alpha = (\alpha_1, \alpha_3, \alpha_5, \alpha_7)$. The parameters α are linearly constrained to respect a minimal and maximal bar thickness

at every vertex.

The target material is East Indian Rosewood, a common material for the production of marimba bars, see Table 5.15. For the initial solution to the non-linear optimization problem, we set $l = 44$ mm, $w = 6.8$ mm, and α_1 to yield a thickness of 2.9 mm. The parameters $\alpha_3, \alpha_5, \alpha_7$ are set to zero. This initial solution yields a pitch frequency of about 500 Hz. We use the objective function

$$g(w, l, \boldsymbol{\alpha}) = (f_1 - 523.3)^2 + \left(\frac{f_2}{4} - 523.3\right)^2,$$

where f_1 denotes the frequency of the pitch and f_2 denotes the frequency of the first overtone of the incumbent solution. Running the optimization routine for this initial solution and a few similar initial solutions reveals that g has many local minima. Thus, this problem cannot be solved with a local optimization strategy alone.

In order to learn more about the nature of the problem, we generate a set of parameter samples that covers the feasible parameter space of $\boldsymbol{\alpha}$. First, we roughly estimate a bounding box of the feasible parameter space as $\alpha_1 \in [0, 1]$, $\alpha_3, \alpha_5, \alpha_7 \in [-1, 1]$. Then we construct the first 10^5 members of the four-dimensional Halton sequence with primes 2,3,5, and 7. The Halton sequence is a low-discrepancy sequence, i.e., a sequence whose elements are uniformly distributed and have a quasi-random appearance, which is commonly used for multi-dimensional Monte Carlo sampling. Of these samples, we only keep the ones that lie in the actual feasible region of $\boldsymbol{\alpha}$ defined by the minimal and maximal bar thickness constraints. This leaves us with about 2,000 samples. For those parameter samples we construct the corresponding marimba bars using the initial values for w and l and evaluate the finite element model to find the pitch and the first overtone. This results in a set of 2,000 frequency pairs, only five of which have a harmonic ratio of 1:4 or higher, all with similar parameter values and similarly shaped marimba bars.

We arbitrarily choose one of these five solutions, which has a harmonic ratio of 1:4.11, as our best guess. Our sampling algorithm had the goal of sampling different harmonic ratios, but it did not consider pitch. Therefore, we obtain a greatly decreased pitch of 153 Hz instead of the desired pitch of 523.3 Hz, although the width and length of the bar are the same as in our initial solution. This reduction in pitch is due to the arched undercut that our current solution possesses. We find that a uniform scaling of all six parameters $w, l, \boldsymbol{\alpha}$ changes all frequencies proportionally, and therefore keeps the harmonic ratio constant. The optimal uniform scaling factor that moves the pitch and the overtone close to the desired values is found by local optimization. This gives us a scaling factor of 0.293 and a solution with a bar length of 12.9 cm, a bar width of 2 cm, a pitch of 522.9 Hz, and a first overtone of 2150 Hz.

Using this as an initial solution for Matlab's local optimization routine `fmincon` is still not good enough to find an optimal solution with a harmonic ratio of 1:4. Therefore we use the Monte Carlo approach to improve the initial solution again. This time we sample the close neighborhood of all six optimization parameters w, l , and $\boldsymbol{\alpha}$. This is achieved by calculating the first 200 members of the six-dimensional Halton sequence with primes 2,3,5,7,11, and 13 and scaling them to the range $[0.99, 1.01]$. A new parameter sample is generated by multiplying the six parameters of the old initial solution with a

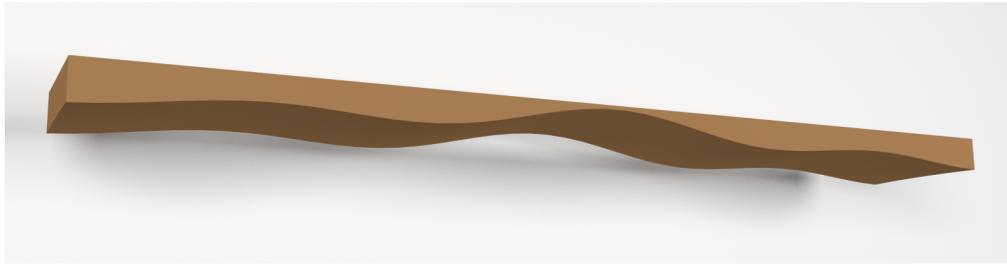


Figure 5.16: The optimal marimba bar with a harmonic ratio of 1:4.

sample from the Halton sequence, therefore inducing parameter perturbations of up to 1%. Out of these 200 new solutions, the best one serves as the new input to `fmincon`. This initial solution is finally good enough to find an optimal solution with a pitch of 523.3 Hz and a first overtone of 2091.5 Hz, which corresponds to a harmonic ratio of 1:3.996. The finalized design, which is illustrated in Fig 5.16, has an arched undercut that is very similar to the ones found in professional instruments.

Possible Optimizations

6.1 Overview

This chapter presents computational results that explore a possibility to speed up the frequency optimization process. In particular, we evaluate the influence of mesh resolution and mesh smoothness on finite element model analysis.

The most important factor that contributes to the duration of a single iteration in the non-linear optimization procedure is the complexity of the finite element mesh that is being evaluated. A finely tessellated mesh and high-order element types lead to a high number of elements and nodes in the mesh. The number of elements directly corresponds to the number of element mass matrices and element stiffness matrices that have to be evaluated. The number of nodes defines the dimensions of the master mass matrix and the master stiffness matrix. Large master matrices lead to longer run-times for the sparse eigenproblem solver.

It is evident that an effective way of speeding up computations is to use coarse finite element meshes. However, the accuracy of the finite element method is highly correlated with the resolution of the mesh. Therefore, it is desirable to use a mesh resolution that is just large enough such that further refinement would not improve the solution significantly. Section 6.2 demonstrates the effects of changes in mesh resolution on the results of finite element analysis.

An additional concern is the loss of geometric detail that may result from a decrease in mesh resolution. If a mesh has high-frequency surface details that can only be represented on a finely tessellated mesh, mesh simplification implicitly performs mesh smoothing by removing those details. Therefore, it is an interesting question whether the presence of fine surface details changes the frequency of natural modes significantly. Section 6.3 performs this analysis by artificially smoothing a mesh with fine surface details and observing that this evokes only small changes in the natural frequencies. The reason that we do not perform mesh simplification in order to study the influence of reduced geometric detail is that this would also result in a decrease of the accuracy of the finite

element model due to the larger size of the elements. If we remove surface details without changing the resolution of the finite element mesh, we can more easily distinguish between the two sources of error.

The combination of these two results justifies simplification of meshes for finite element analysis even if this removes geometric surface details. A mesh simplification routine could be incorporated into our frequency optimization framework by starting analysis on a coarse mesh and gradually moving to a finer mesh as the optimization routine approaches a local optimum to ensure correctness. The actual implementation of this optimization strategy is outside the scope of this work.

6.2 Mesh-Resolution Benchmarks

The accuracy of finite element solutions depends on three major factors: the order of individual elements, the degree of distortion of individual elements, and the resolution of the finite element mesh. The order of an element is the degree of polynomial in the primary variable, i.e., the displacement \mathbf{u} , that can be represented exactly on the element domain. Linear elements, such as the linear tetrahedral element, are the lowest-order elements that are commonly used. They can only exactly represent linear functions in \mathbf{u} , and one of their uses is to predict the elongation in a prismatic bar element. Quadratic elements, such as the wedge element or the quadratic tetrahedral element, are sufficient for most other scenarios in solid mechanics such as the prediction of bending phenomena. The degree of distortion refers to the deviation of the shape of an element domain from its standard domain in natural coordinates. The negative effect of strongly distorted elements on solution accuracy is explored by Lee and Bathe [LB93].

In this section, we present benchmarks to evaluate the dependence of solution accuracy on the resolution of the finite element mesh. In particular, we want to analyze at which mesh resolution the result is sufficiently converged, such that further refinement no longer improves the solution significantly. One benchmark analyzes the effects of mesh resolution on a thin rectangular plate and the other on a hemispherical shell. We choose these two geometries because they can be parametrized easily w.r.t. mesh resolution, and they are prominent in the design of struck idiophones.

Plate Benchmark. The first benchmark evaluates the natural frequencies of a thin rectangular plate with the dimensions of plate 2 in Section 5.1. We generate ten regular grids on the surface of the plate. The coarsest grid has a cell side length of 10 mm, and the finest grid has a cell side length of 1 mm. This yields 36 cells for the coarsest and 2331 cells for the finest grid. Fig. 6.1 plots the relative deviations of the first twelve natural frequencies w.r.t. to the result obtained from the most finely resolved mesh with the respective finite element model. The wedge model and the quadratic tet model behave similarly in that the deviations in frequency decrease almost monotonically as the grid resolution is increased, but the wedge model converges slightly faster. E.g., for a cell side length of 2.8 mm the maximal deviation from the final result is 0.6% in the wedge model and 1.1% in the quadratic tet model. In general, the changes in frequency are greater

for higher-frequency modes than for low-frequency modes. The reason is that the mode shapes for high-frequency modes have steeper stress gradients that cannot be captured accurately on a low-resolution finite element mesh.

The frequency development for the thin-shell model is much more chaotic than for the other models. The natural frequencies only start to stabilize at a cell side length of 1.7 mm. This explains the comparatively bad predictive power of the thin-shell model in the evaluation of the aluminium plates in Section 5.1. However, the wedge and quadratic tet model only outperform the thin-shell model strongly on a few modes, namely modes 6, 8, and 11. The remaining modes behave stably for cell side lengths of 2.8 mm and smaller. Most notably the frequency of the pitch itself retains its value on every grid with a maximal deviation of 0.3%, compared to 0.4% for the wedge model and 0.6% for the quadratic tet model.

Shell Benchmark. The second benchmark evaluates the natural frequencies of a hemispherical shell with a radius of 5 cm and a thickness of 4 mm. The shell is approximated with five triangle meshes at different resolutions. Each mesh is generated by subdividing an icosahedron and removing the lower half of the mesh. With this procedure, the triangle edge length is halved for each subsequent mesh, and the number of elements is quadrupled. The coarsest mesh has edge lengths of 29.1 mm, and the finest mesh has edge lengths of 1.9 mm. This yields 40 triangles for the coarsest and 10240 triangles for the finest mesh. Once more, convergence behavior is observed by comparing results to the frequencies computed on the finest mesh.

Fig. 6.2 plots the relative deviations of the first twelve natural frequencies for the three finite element models. The wedge model and the quadratic tet model show a behavior similar to the first benchmark. The deviations decrease monotonically, but the wedge model converges slightly faster. At the 7.5 mm benchmark, the maximal deviation for the wedge model is 1.2%, and the maximal deviation for the quadratic tet model is 2.0%. The thin-shell model performs significantly worse for all modes except 1 and 2. Most of the other modes start to converge at the 7.5 mm mark, except for modes 7, 9, and 12. Comparing the results of the 1.9 mm benchmark for the thin-shell model and the wedge model reveals that they converge to slightly differently results. The frequency of the pitch differs by 1.6%, and the frequency of higher modes differs by up to 4.4%.

6.3 Mesh-Smoothness Benchmark

The local optimization routine introduced in Section 4.4, as well as the global optimization performed in Section 5.2.3, spend most of their computation time on building the element matrices of the finite element model and performing sparse eigendecomposition. To compute the master stiffness matrix for the rabbit bell model that was used in Section 5.2.2, a total of 8478 element stiffness matrices of dimensions 45-by-45 are computed by eight-point Gauss quadrature. This computation, combined with the evaluation of the mass matrix, takes 7 seconds on an Intel Core i7-4700HQ CPU using the OpenFEM library. Every time the optimization routine evaluates the objective function, this computation

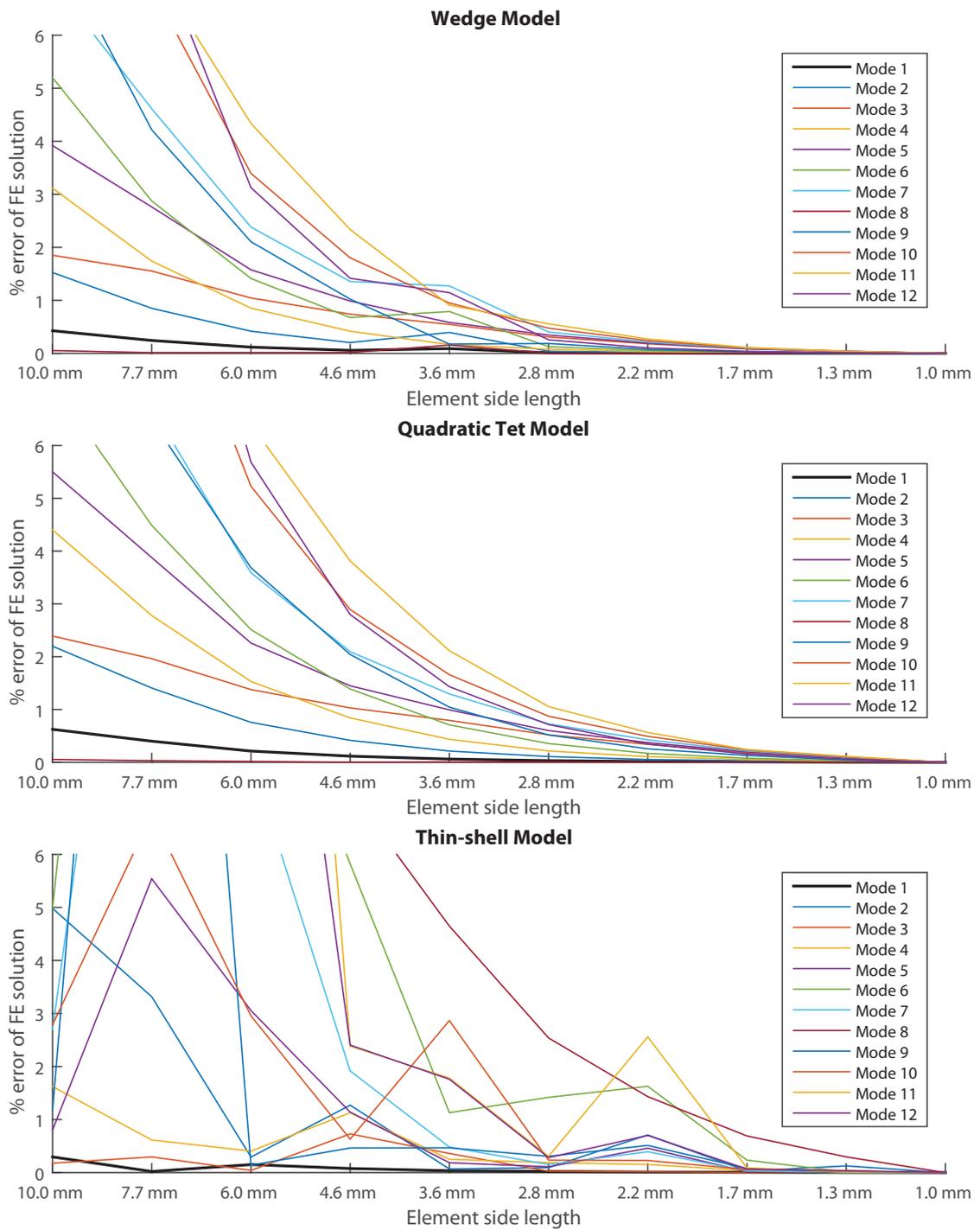


Figure 6.1: Comparison of finite element solutions for different finite element meshes of a thin rectangular plate.

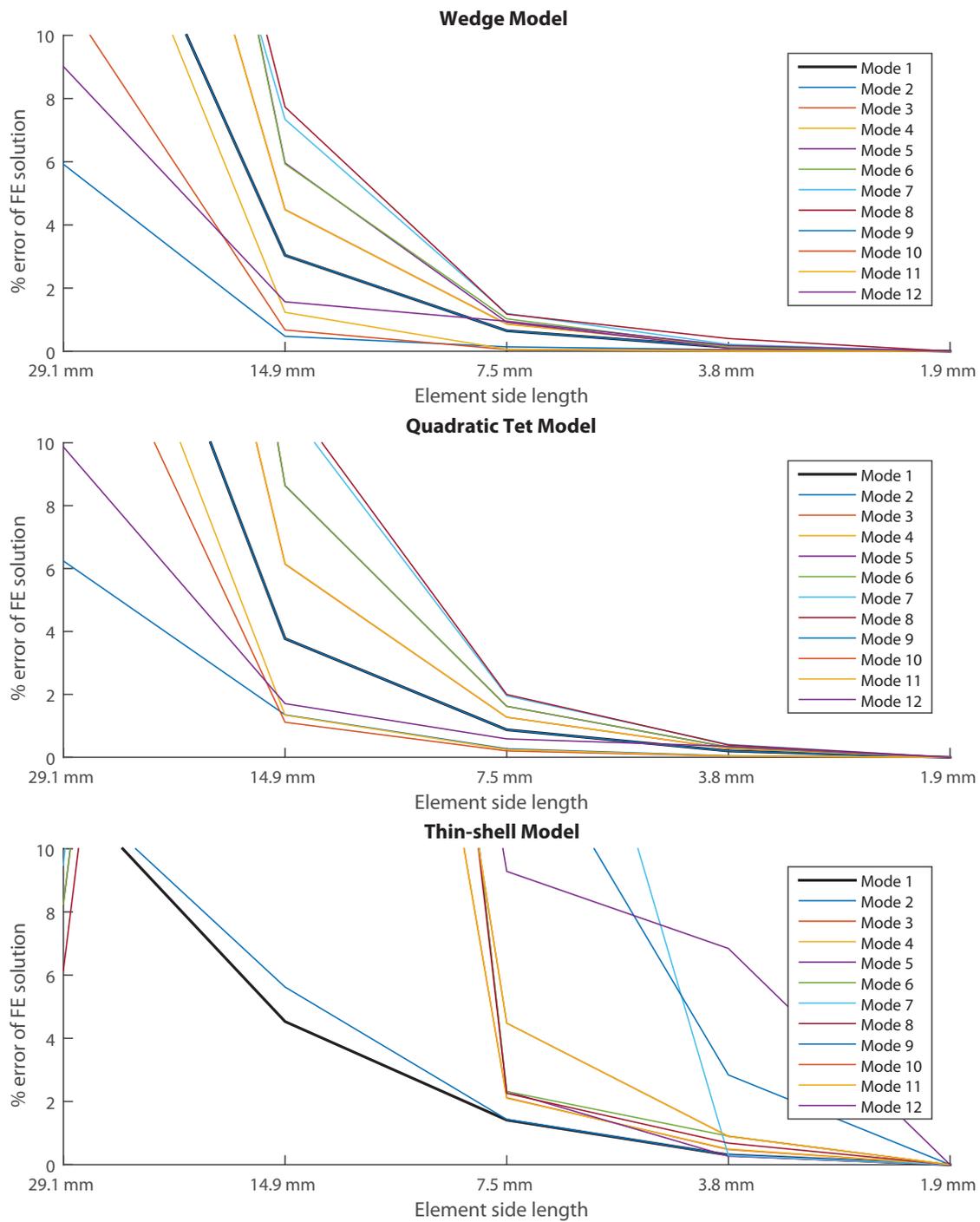


Figure 6.2: Comparison of finite element solutions for different finite element meshes of a hemispherical shell.

is performed. Since the gradient of the objective function is computed numerically, the number of objective function evaluations increases drastically with the number of optimization parameters.

The number of rows and columns of the master stiffness matrix is dictated by the number of degrees of freedom in the finite element mesh. For the rabbit mesh, the master stiffness matrix has the dimensions 12852-by-12852. Since the matrix is sparse, we use Matlab’s function `eigs` to find the frequencies of the system. This algorithm is linear in the number of non-zero entries in \mathbf{K} and in the number of eigenvalues that have to be found. Determining the frequency of the pitch alone means finding the seven lowest eigenvalues because the eigenvalue zero is found six times. This computation takes 11 seconds for the rabbit model.

For smooth geometries, the finite element mesh can be simplified while maintaining an accurate description of the geometry. If care is exercised when sizing the elements, the wedge model and the quadratic tet model yield good results with a greatly reduced mesh resolution, as discussed in Section 6.2. However, geometric information is lost through mesh simplification if the surface has high-frequency details such as small bumps. In this case, mesh simplification indirectly performs mesh smoothing.

This section evaluates how mesh smoothing affects the natural frequencies of a thin-shelled solid. To implement mesh smoothing, we use the spectral technique reviewed by Lévy et al. [LZ10]. This technique is based on the eigendecomposition

$$\mathbf{L} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$$

of the symmetric Laplacian that was introduced in Section 4.2.4. To define the manifold harmonics parametrization, we used k eigenvectors corresponding to the k smallest eigenvalues to parametrize a set of offset vectors. However, \mathbf{Q} also provides a suitable basis to represent the geometry of the underlying mesh itself. Let $\mathbf{V} = [\mathbf{x}|\mathbf{y}|\mathbf{z}]$ denote the n -by-3 matrix that lists the x -, y -, and z -coordinates of the vertices. The vertex positions can be projected onto the orthonormal basis \mathbf{Q} by computing

$$\tilde{\mathbf{V}} = [\tilde{\mathbf{x}}|\tilde{\mathbf{y}}|\tilde{\mathbf{z}}] = \mathbf{Q}^{-1}\mathbf{V} = \mathbf{Q}^T\mathbf{V}.$$

The rows of matrix $\tilde{\mathbf{V}}$ are the coefficients of the individual eigenvectors needed to reconstruct the vertex positions. For most triangle meshes, the top rows in $\tilde{\mathbf{V}}$ carry the most important information because they determine the low-frequency characteristics of the mesh. The coefficients in the bottom rows encode high-frequency surface variations. By scaling the entries of $\tilde{\mathbf{V}}$, certain frequency bands can be amplified or damped. For mesh smoothing purposes, it is useful to leave the low-frequency coefficients as they are and to decrease or zero out high-frequency coefficients.

Our test mesh is an icosphere with 2,562 vertices and 5,120 faces. We add layers of sinusoidal noise at different frequencies to the vertex positions, such that the sphere develops bumps on the surface. Starting from this mesh, we create a series of smoothed meshes with the spectral method described above. The plots in Fig. 6.3 illustrate how the the spectral coefficients are damped to achieve different degrees of smoothness. Note that

the coefficients of the first two eigenvectors are left untouched in order to prevent global shrinking of the mesh. The remaining coefficients are scaled with the transformation

$$\tilde{x}_{i,m} = \tilde{x}_i \cdot \left(\frac{n-i}{n} \right)^{2m^{1.3}},$$

where $\tilde{x}_{i,m}$ is the x-coefficient of the i th eigenvector for the mesh of smoothness level m . If $m = 0$, the coefficients are not changed, and the larger m becomes, the more strongly the coefficients are damped. The y-, and z-coefficients are scaled analogously.

From each of the ten meshes, a mean-curvature skeleton is computed in order to create an inner offset surface. The volume between the outer and the inner surface describes a spherical thin-shelled solid. Fig. 6.4 plots the frequencies of modes one to twelve for the ten different geometries as calculated by the wedge finite element model. It is evident that the removal of high-frequency surface details only has a minor influence on the natural frequencies. The frequency of the pitch for $m = 0$ deviates only by 3.6% from the pitch for $m = 9$. For higher modes, this deviation stays below 6%.

This result shows that it is possible to achieve meaningful results if frequency analysis is performed on simplified meshes that miss fine surface details. Combined with the knowledge that good results can be obtained even for coarse finite element meshes, this may allow shorter computation times even for complex models.

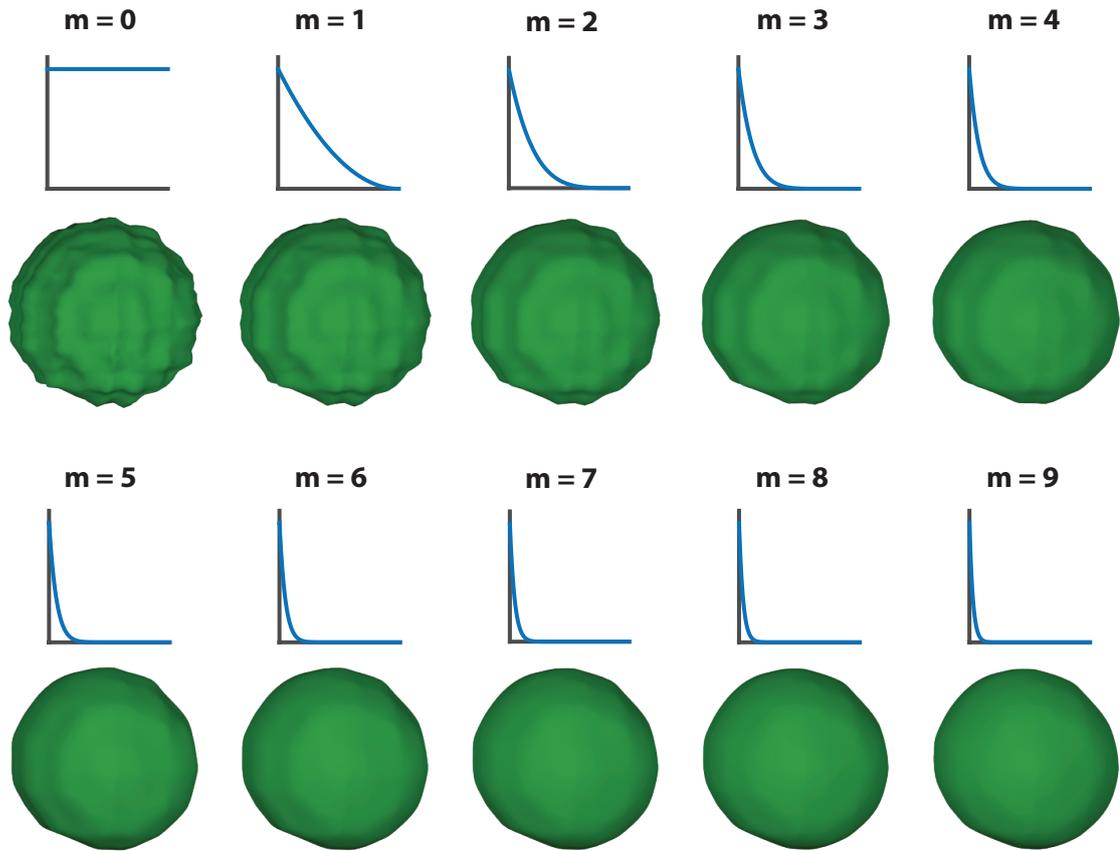


Figure 6.3: The line plots show the scaling factors of the eigenvector coefficients for different values of m . The abscissa represents the index of the eigenvector, and the ordinate the scaling factor between 0 and 1. Below the line plots, the resulting meshes are illustrated.

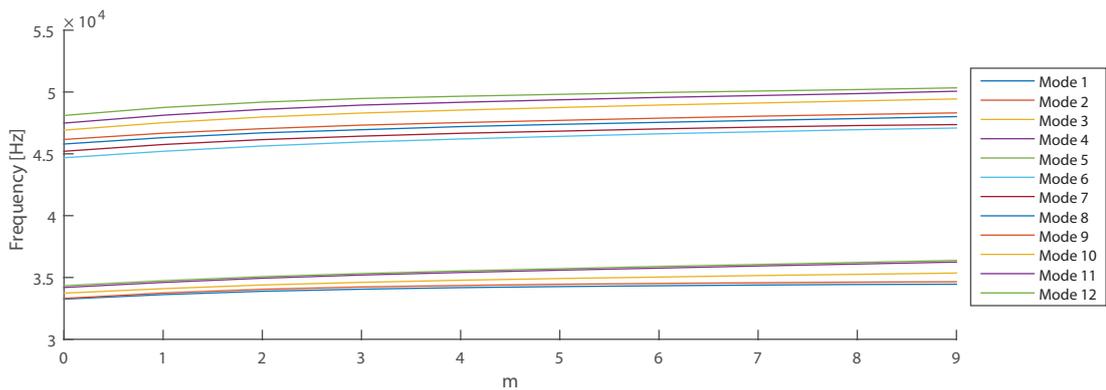


Figure 6.4: Frequencies of modes one to twelve of the spherical thin-shelled solids. The abscissa represents the smoothness level m .

Conclusion

7.1 Synopsis

This thesis explores a new application of shape optimization for the fabrication of physical artifacts. We combine computational modal analysis with a non-linear optimization routine in order to synthesize solids with desired sound properties. The main goal is to optimize the frequency of the pitch, which is the most distinctive property of the sounds exhibited by a musical instrument.

Our first contribution is a number of suitable shape parametrizations that enable the description of instrument geometries. Motivated by the observation that many struck idiophones consist of thin plates, e.g., xylophones and marimbas, and thin shells, e.g., bells and cymbals, we choose an offset surface representation. The solid geometry is defined as the volume delimited by two surfaces, one of which remains fixed, and the other is displaced along a set of offset vectors to achieve the desired optimization goals. We provide three parametrizations of the offset vector magnitudes that provide different levels of control over the instrument shape and different capacities to handle complex geometries. The simplest parametrization offsets the outer surface along the per-vertex normals at a distance that is constant across the mesh. This approach can only be applied to simple, low-curvature shapes like bells because it is liable to produce self-intersections of the inner surface for more complex geometries. The second parametrization variant resolves this issue by replacing the per-vertex normals with offset vectors that point in the direction of a meso-skeleton located at the center of the object. By displacing the vertices of the outer surface along these offset directions, self-intersections are prevented for all but very jaggy geometries. Like for the first parametrization, a single parameter is used to control the offset magnitudes across the mesh, but the magnitudes are limited to never exceed the distance to the meso-skeleton. The third parametrization introduces an adjustable number of parameters in order to handle objectives that go beyond optimizing the frequency of the pitch. This is achieved by representing offset magnitudes as weighted sums of eigenfunctions of the mesh Laplacian, where the weights serve as optimization

parameters. An application of this technique is the optimization of harmonic overtone ratios, which is an important quality criterion of professional instruments.

In order to evaluate the pitch and overtone frequencies of a geometry given as the volume between two surfaces, we use finite element modal analysis. The choice of element has a great impact on solution accuracy, and thus we develop different elements and evaluate their predictive power using a set of thin aluminium plates. We conclude that linear elements like the linear tetrahedron are not suited to determine natural frequencies because they cannot represent stress distributions with sufficient accuracy. Instead we opt for quadratic elements like the quadratic tetrahedron and the 15-node triangular wedge element because they produce results that are consistent with our measurements. Using only reference material parameters we achieve a median error of 1.7% with the wedge model across the first 24 natural frequencies of six aluminium plates. To increase accuracy, we perform material parameter estimation on a fraction of the measured frequencies. Using these optimized parameters in the finite element model, we can reduce the median errors by up to 70%.

We incorporate finite element modal analysis into an optimization routine in order to synthesize instrument geometries that exhibit the desired pitch. Using the basic per-vertex-normal parametrization, we optimize a hemiellipsoidal bell to exhibit a pitch frequency of 1760 Hz. In order to produce it, we build a two-part mold from oiled molding sand and cast the bell from molten tin. The measured frequency deviates from the predicted frequency by less than 3%. Our second physical result is a bell in the shape of a rabbit that is optimized using the skeleton-parametrization method. The resulting solid has undercuts and can therefore not be produced using sand molds. Instead we opt for molds from natural rubber, which is flexible enough to describe concave features. For this bell we experience an initial frequency error of 11%, but through material parameter estimation we can reduce the error to 6%. The most likely explanation for the remaining error is that our crude fabrication pipeline introduces manufacturing flaws that distort the pitch.

The utility of the manifold harmonics parametrization is demonstrated on the example of a marimba bar. Professional marimba bars are tuned by adding an arched carving to the bottom in order to achieve the harmonic ratio 1:4 between the pitch and the first overtone. We compute a bar shape that exhibits the same harmonic ratio through parametrization of the offset surface using four Laplacian basis functions. To find a good initial solution, we perform Monte Carlo sampling of the parameter space and finally compute an optimal solution using Matlab's non-linear local optimization routine. The resulting shape reproduces the characteristic design of real marimba bars with a wide central undercut.

In the last part of the thesis, we explore the influence of finite element mesh resolution on the natural frequency spectrum. We present a number of benchmarks that evaluate the natural frequencies of objects at different mesh resolutions. The conclusion is that the wedge model and the quadratic tetrahedral model yield consistent results even for coarse meshes, but the behavior of the thin-shell model is less predictable, and finer meshes are required for reliable results. Another benchmark examines the effect of fine surface

details on natural frequencies. Using Laplacian smoothing in the spectral domain, we progressively filter high-frequency components of a thin-shelled object and observe the effect on the frequency spectrum. The results suggest that the influence of high-frequency surface details is very limited, and that coarse mesh approximations can be used to speed up finite element modal analysis.

7.2 Future Work

A key limitation of our method is the inability to purposefully influence the frequency of a specific mode while keeping the others constant due to the global influence of manifold harmonics basis functions. This is why Monte Carlo techniques need to be employed in order to optimize harmonic ratios between overtones. Literature on musical instruments suggests that targeting specific modes through geometric variations is difficult in general. However, instrument makers have developed techniques to tune instruments such as bells by removing material from particular locations. A natural frequency is influenced most by notches that are placed where the corresponding mode shape has its largest deflection. An interesting research direction might be to evaluate this relationship more rigorously in order to exert better control over particular natural modes.

This may lead to shape parametrization methods that are better suited to the optimization of harmonic ratios than the manifold harmonics basis. The problem with the latter approach is that Laplacian basis functions change the shape globally, and thus the relationship between function coefficients and natural frequencies is very obscure. Using parameters that modify the shape locally may yield objective functions with fewer local minima, thereby reducing the need for global optimization.

We also see room for improvement regarding the run-time performance of our finite element computations. The knowledge that coarse finite element meshes and smooth geometries can yield meaningful results might make it possible to incorporate an automatic mesh simplification routine into the optimization pipeline. This could enable faster convergence to an approximate solution on a coarse mesh and refinement of the solution on a detailed mesh.

Bibliography

- [ACOL00] Marc Alexa, Daniel Cohen-Or, and David Levin. As-rigid-as-possible shape interpolation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 157–164. ACM Press/Addison-Wesley Publishing Co., 2000.
- [Bat06] K.J. Bathe. *Finite Element Procedures*. Prentice Hall, 2006.
- [BBJP12] Moritz Bächer, Bernd Bickel, Doug L. James, and Hanspeter Pfister. Fabricating articulated characters from skinned meshes. *ACM Trans. Graph.*, 31(4):47:1–47:9, July 2012.
- [BBO⁺10] Bernd Bickel, Moritz Bächer, Miguel A Otaduy, Hyunho Richard Lee, Hanspeter Pfister, Markus Gross, and Wojciech Matusik. Design and fabrication of materials with desired deformation behavior. *ACM Transactions on Graphics (TOG)*, 29(4):63, 2010.
- [BCT⁺99] Ingolf Bork, Antoine Chaigne, Louis-Cyrille Trebuchet, Markus Kosfelder, and David Pillot. Comparison between modal analysis and finite element modelling of a marimba bar. *Acta Acustica united with Acustica*, 85(2):258–266, 1999.
- [BDT⁺08] Nicolas Bonneel, George Drettakis, Nicolas Tsingos, Isabelle Viaud-Delmon, and Doug James. Fast modal sounds with scalable frequency-domain synthesis. *ACM Trans. Graph.*, 27(3):24:1–24:9, August 2008.
- [BSM99a] J Bretos, C Santamaria, and J Alonso Moral. 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.
- [BSM99b] J Bretos, C Santamaria, and J Alonso Moral. Vibrational patterns and frequency responses of the free plates and box of a violin obtained by finite element analysis. *The Journal of the Acoustical Society of America*, 105(3):1942–1950, 1999.

- [BWBSH14] Moritz Bächer, Emily Whiting, Bernd Bickel, and Olga Sorkine-Hornung. Spin-it: Optimizing moment of inertia for spinnable objects. *ACM Trans. Graph.*, 33(4):96:1–96:10, July 2014.
- [CAJ09] Jeffrey N. Chadwick, Steven S. An, and Doug L. James. Harmonic shells: A practical nonlinear sound model for near-rigid thin shells. *ACM Trans. Graph.*, 28(5):119:1–119:10, December 2009.
- [Can86] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pages 679–698, 1986.
- [CCA⁺12] Jacques Calì, Dan A. Calian, Cristina Amati, Rebecca Kleinberger, Anthony Steed, Jan Kautz, and Tim Weyrich. 3d-printing of non-assembly, articulated models. *ACM Trans. Graph.*, 31(6):130:1–130:8, November 2012.
- [CLM⁺13] Duygu Ceylan, Wilmot Li, Niloy J. Mitra, Maneesh Agrawala, and Mark Pauly. Designing and fabricating mechanical automata from mocap sequences. *ACM Trans. Graph.*, 32(6):186:1–186:11, November 2013.
- [CSCR09] Ioan Curtu, M Stanciu, N Cretu, and I Rosca. Modal analysis of different types of classical guitar bodies. In *Proceedings of the 10th WSEAS International Conference on Acoustics & Music: Theory & Applications-AMTA09 (ISTP/ISI Proceeding of Thomson Scientific-Institute for Scientific Information)*, pages 23–25, 2009.
- [CTN⁺13] Stelian Coros, Bernhard Thomaszewski, Gioacchino Noris, Shinjiro Sueda, Moira Forberg, Robert W. Sumner, Wojciech Matusik, and Bernd Bickel. Computational design of mechanical characters. *ACM Trans. Graph.*, 32(4):83:1–83:12, July 2013.
- [CZJ12] Jeffrey N. Chadwick, Changxi Zheng, and Doug L. James. Precomputed acceleration noise for improved rigid-body sound. *ACM Trans. Graph.*, 31(4):103:1–103:9, July 2012.
- [CZXX14] Xiang Chen, Changxi Zheng, Weiwei Xu, and Kun Zhou. An asymptotic numerical method for inverse elastic shape design. *ACM Transactions on Graphics (TOG)*, 33(4):95, 2014.
- [DHL14] Jérémie Dumas, Jean Hergel, and Sylvain Lefebvre. Bridging the gap: Automated steady scaffoldings for 3d printing. *ACM Trans. Graph.*, 33(4):98:1–98:10, July 2014.
- [DLVC99] Julien Dompierre, Paul Labbé, Marie-Gabrielle Vallet, and Ricardo Camarero. How to subdivide pyramids, prisms and hexahedra into tetrahedra. In *Proceedings of the 8th International Meshing Roundtable*, 1999.

- [DMA02] Mathieu Desbrun, Mark Meyer, and Pierre Alliez. Intrinsic parameterizations of surface meshes. In *Computer Graphics Forum*, volume 21, pages 209–218. Blackwell Publishing, Inc, 2002.
- [DMSB99] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324. ACM Press/Addison-Wesley Publishing Co., 1999.
- [Fel13] Carlos A. Felippa. Advanced finite element methods for solids, plates and shells. <http://www.colorado.edu/engineering/CAS/courses.d/AFEM.d/Home.html>, May 2013. Lecture Notes.
- [Fel14] Carlos A. Felippa. Introduction to finite element methods. <http://www.colorado.edu/engineering/CAS/courses.d/IFEM.d/Home.html>, December 2014. Lecture Notes.
- [FR98] N.H. Fletcher and T.D. Rossing. *The Physics of Musical Instruments*. Springer, 1998.
- [Gav93] William W. Gaver. Synthesizing auditory icons. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pages 228–235, New York, NY, USA, 1993. ACM.
- [GPSG11] Gustavo Paulinelli Guimarães, Rogério Pirk, Carlos d’Andrade Souto, and Luiz Carlos S. Góes. Acoustic modal analysis of cylindrical-type cavities. In *Proceedings of the 8th International Conference on Structural Dynamics*, 2011.
- [IMH05] Takeo Igarashi, Tomer Moscovich, and John F Hughes. As-rigid-as-possible shape manipulation. *ACM transactions on Graphics (TOG)*, 24(3):1134–1141, 2005.
- [JBP06] Doug L. James, Jernej Barbič, and Dinesh K. Pai. Precomputed acoustic transfer: Output-sensitive, accurate sound generation for geometrically complex vibration sources. *ACM Trans. Graph.*, 25(3):987–995, July 2006.
- [LAJJ14] Timothy R. Langlois, Steven S. An, Kelvin K. Jin, and Doug L. James. Eigenmode compression for modal sound models. *ACM Trans. Graph.*, 33(4):40:1–40:9, July 2014.
- [LB93] Nam-Sua Lee and Klaus-Jurgen Bathe. Effects of element distortions on the performance of isoparametric elements. *International Journal for Numerical Methods in Engineering*, 36(20):3553–3576, 1993.

- [Lév06] Bruno Lévy. Laplace-beltrami eigenfunctions: towards an algorithm that “understands” geometry. In *Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on*, pages 13–13. IEEE, 2006.
- [LJ14] Timothy R. Langlois and Doug L. James. Inverse-foley animation: Synchronizing rigid-body motions to sound. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2014)*, 33(4), August 2014.
- [LPRM02] Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérôme Maillot. Least squares conformal maps for automatic texture atlas generation. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 362–371. ACM, 2002.
- [LSZ⁺14] Lin Lu, Andrei Sharf, Haisen Zhao, Yuan Wei, Qingnan Fan, Xuelin Chen, Yann Savoye, Changhe Tu, Daniel Cohen-Or, and Baoquan Chen. Build-to-last: Strength to weight 3d printed objects. *ACM Trans. Graph.*, 33(4):97:1–97:10, July 2014.
- [Lu13] Ye Lu. *Comparison of finite element method and modal analysis of violin top plate*. PhD thesis, McGill University, 2013.
- [LZ10] Bruno Lévy and Hao (Richard) Zhang. Spectral mesh processing. In *ACM SIGGRAPH 2010 Courses*, SIGGRAPH ’10, pages 8:1–8:312, New York, NY, USA, 2010. ACM.
- [MAB⁺15] Przemyslaw Musialski, Thomas Auzinger, Michael Birsak, Michael Wimmer, and Leif Kobbelt. Reduced-order shape optimization using offset surfaces. *ACM Transactions on Graphics (ACM SIGGRAPH 2015)*, 34(4):to appear–9, August 2015.
- [MDSB03] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, 2003.
- [MEH12] H.A. Mang, J. Eberhardsteiner, and G. Hofstetter. *Festigkeitslehre*. Springer-Lehrbuch Technik. Springer Vienna, 2012.
- [NISA06] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Laplacian mesh optimization. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 381–389. ACM, 2006.
- [OCE01] James F. O’Brien, Perry R. Cook, and Georg Essl. Synthesizing sounds from physically based motion. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’01, pages 529–536, New York, NY, USA, 2001. ACM.

- [OH99] James F. O’Brien and Jessica K. Hodgins. Graphical modeling and animation of brittle fracture. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’99, pages 137–146, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [OSG02] James F. O’Brien, Chen Shen, and Christine M. Gatchalian. Synthesizing sounds from rigid-body simulations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA ’02, pages 175–181, New York, NY, USA, 2002. ACM.
- [PDJ⁺01] Dinesh K. Pai, Kees van den Doel, Doug L. James, Jochen Lang, John E. Lloyd, Joshua L. Richmond, and Som H. Yau. Scanning physical interaction behavior of 3d objects. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’01, pages 87–96, New York, NY, USA, 2001. ACM.
- [Pet10] M. Petyt. *Introduction to Finite Element Vibration Analysis*. Cambridge University Press, 2010.
- [PW89] A. Pentland and J. Williams. Good vibrations: Modal dynamics for graphics and animation. *SIGGRAPH Comput. Graph.*, 23(3):207–214, July 1989.
- [PWLSH13] Romain Prévost, Emily Whiting, Sylvain Lefebvre, and Olga Sorkine-Hornung. Make it stand: Balancing shapes for 3d fabrication. *ACM Trans. Graph.*, 32(4):81:1–81:10, July 2013.
- [RL06] Nikunj Raghuvanshi and Ming C. Lin. Interactive sound synthesis for large scale environments. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games*, I3D ’06, pages 101–108, New York, NY, USA, 2006. ACM.
- [RSM⁺10] Nikunj Raghuvanshi, John Snyder, Ravish Mehra, Ming Lin, and Naga Govindaraju. Precomputed wave simulation for real-time sound propagation of dynamic sources in complex scenes. *ACM Trans. Graph.*, 29(4):68:1–68:11, July 2010.
- [RYL10] Zhimin Ren, Hengchin Yeh, and Ming C Lin. Synthesizing contact sounds between textured models. In *Virtual Reality Conference (VR), 2010 IEEE*, pages 139–146. IEEE, 2010.
- [SCOL⁺04] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184. ACM, 2004.

- [SLMB05] Alla Sheffer, Bruno Lévy, Maxim Mogilnitsky, and Alexander Bogomyakov. Abf++: fast and robust angle based flattening. *ACM Transactions on Graphics (TOG)*, 24(2):311–330, 2005.
- [STBG12] Mélina Skouras, Bernhard Thomaszewski, Bernd Bickel, and Markus Gross. Computational design of rubber balloons. In *Computer Graphics Forum*, volume 31, pages 835–844. Wiley Online Library, 2012.
- [STC⁺13] Mélina Skouras, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, and Markus Gross. Computational design of actuated deformable characters. *ACM Trans. Graph.*, 32(4):82:1–82:10, July 2013.
- [STK⁺14] Mélina Skouras, Bernhard Thomaszewski, Peter Kaufmann, Akash Garg, Bernd Bickel, Eitan Grinspun, and Markus Gross. Designing inflatable structures. *ACM Transactions on Graphics (TOG)*, 33(4):63, 2014.
- [SVB⁺12] Ondrej Stava, Juraj Vanek, Bedrich Benes, Nathan Carr, and Radomír Měch. Stress relief: Improving structural strength of 3d printable objects. *ACM Trans. Graph.*, 31(4):48:1–48:11, July 2012.
- [TAOZ12] Andrea Tagliasacchi, Ibraheem Alhashim, Matt Olson, and Hao Zhang. Mean curvature skeletons. *Comp. Graph. Forum*, 31(5):1735–1744, August 2012.
- [Tau95] Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358. ACM, 1995.
- [TCG⁺14] Bernhard Thomaszewski, Stelian Coros, Damien Gauge, Vittorio Megaro, Eitan Grinspun, and Markus Gross. Computational design of linkage-based characters. *ACM Trans. Graph.*, 33(4):64:1–64:9, July 2014.
- [TG69] S. Timoshenko and J.N. Goodier. *Theory of elasticity*. McGraw-Hill classic textbook reissue series. McGraw-Hill, 1969.
- [TH92] Tapio Takala and James Hahn. Sound rendering. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '92*, pages 211–220, New York, NY, USA, 1992. ACM.
- [UMI10] Nobuyuki Umetani, Jun Mitani, and Takeo Igarashi. Designing custom-made metallophone with concurrent eigenanalysis. In *Proceedings of the Conference on New Interfaces for Musical Expression (NIME)*, 2010.
- [vdDKP01] Kees van den Doel, Paul G. Kry, and Dinesh K. Pai. Foleyautomatic: Physically-based sound effects for interactive simulation and animation. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '01*, pages 537–544, New York, NY, USA, 2001. ACM.

- [vdDKP04] Kees van den Doel, Dave Knott, and Dinesh K. Pai. Interactive simulation of complex audiovisual scenes. *Presence: Teleoper. Virtual Environ.*, 13(1):99–111, February 2004.
- [vdDP96] Kees van den Doel and Dinesh K. Pai. Synthesis of shape dependent sounds with physical modeling. In *Proceedings of the International Conference on Auditory Displays*, 1996.
- [vdDP98] Kees van den Doel and Dinesh K. Pai. The sounds of physical shapes. *Presence: Teleoperators and Virtual Environments*, 7(4):382–395, 1998.
- [VL08] Bruno Vallet and Bruno Lévy. Spectral geometry processing with manifold harmonics. In *Computer Graphics Forum*, volume 27, pages 251–260. Blackwell Publishing, Inc, 2008.
- [WWY⁺13] Weiming Wang, Tuanfeng Y. Wang, Zhouwang Yang, Ligang Liu, Xin Tong, Weihua Tong, Jiansong Deng, Falai Chen, and Xiuping Liu. Cost-effective printing of 3d objects with skin-frame structures. *ACM Trans. Graph.*, 32(6):177:1–177:10, November 2013.
- [YJKK10] Yonggyun Yu, In Gwun Jang, In Kyum Kim, and Byung Man Kwak. Nodal line optimization and its application to violin top plate design. *Journal of Sound and Vibration*, 329(22):4785–4796, 2010.
- [ZJ10] Changxi Zheng and Doug L. James. Rigid-body fracture sound with pre-computed soundbanks. *ACM Trans. Graph.*, 29(4):69:1–69:13, July 2010.
- [ZJ11] Changxi Zheng and Doug L. James. Toward high-quality modal contact sound. *ACM Trans. Graph.*, 30(4):38:1–38:12, July 2011.
- [ZLS07] Rhaleb Zayer, Bruno Lévy, and Hans-Peter Seidel. Linear angle based parameterization. In *Fifth Eurographics Symposium on Geometry Processing-SGP 2007*, pages 135–141. Eurographics Association, 2007.
- [ZPZ13] Qingnan Zhou, Julian Panetta, and Denis Zorin. Worst-case structural analysis. *ACM Trans. Graph.*, 32(4):137:1–137:12, July 2013.