Parameter Free Optimization of Shape Adaptive Shell Structures

Helmut Masching

Lehrstuhl für Statik
Prof. Dr.-Ing. Kai-Uwe Bletzinger
Technische Universität München

Schriftenreihe des
Lehrstuhls für Statik der
Technischen Universität München

Band 31

Helmut Masching

Parameter Free Optimization of Shape Adaptive Shell Structures

München 2016

Veröffentlicht durch

Kai-Uwe Bletzinger
Lehrstuhl für Statik
Technische Universität München
Arcisstraße 21 80333 München

Tel.: +49(0)89 289 22422
Fax: +49(0)89 289 22421
E-Mail: kub@tum.de
Internet: www.st.bgu.tum.de

ISBN: 978-3-943683-37-0

TECHNISCHE UNIVERSITÄT MÜNCHEN

Ingenieurfakultät Bau Geo Umwelt

Lehrstuhl für Statik

Parameter Free Optimization of Shape Adaptive Shell Structures

Helmut Masching

Vollständiger Abdruck der von der Fakultät für Bau Geo Umwelt der Technischen Universität München zur Erlangung des akademischen Titels eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:
Prof. Dr.-Ing. habil. Fabian Duddeck

Prüfer der Dissertation:

1. Prof. Dr.-Ing. Kai-Uwe Bletzinger
2. Prof. Dr.-Ing. Axel Schumacher, Bergische Universität Wuppertal
3. Prof. Dr.-Ing. Horst Baier

Die Dissertation wurde am 15.09.2015 an der Technischen Universität München eingereicht und durch die Ingenieurfakultät Bau Geo Umwelt am 05.04.2016 angenommen.

Abstract

This thesis combines structural optimization with the field of adaptive structures. While structural optimization is well established in classical engineering in the meantime, only few optimization is done concerning adaptive or smart structures. Smart structures are used in order to enable more efficient and powerful structural designs. This increase in efficiency usually is achieved by applying control and actuation systems, which influence the structure selectively by additional loading or by modifying the structure's boundary conditions. Goal of this thesis is to combine these two topics, in order to obtain best controllability of structures.

For this purpose, mechanisms allowing a most efficient deformation of thin shell structures are discussed. These mechanisms are generated using parameter free optimization, whereat the influence of different optimization setups is discussed. Furthermore, an enlarged design control concept is presented, meeting the concerns and additional challenges of optimizing an adapted structure. As an example of use, an experimental intelligent airfoil is considered.

In the last part, the capabilities of structural optimization in order to generate bistable structures are investigated. In the field of bistable structures, mainly shell structures with constant curvature and uniform material distribution and orientation are considered so far. In this thesis, methods of free shape and material optimization are applied in order to generate novel bistable shell structures.

Zusammenfassung

Diese Arbeit befasst sich mit dem Zusammenführen der Disziplin der Strukturoptimierung mit dem Themenfeld adaptiver Strukturen. Während die Strukturoptimierung in den klassischen Bereichen des Ingenieurswesens mittlerweile etabliert ist, wird im Themengebiet der adaptiven oder intelligenten Strukturen dagegen wenig Optimierung betrieben. Intelligente Strukturen werden eingesetzt, um noch effizientere und leistungsfähigere Strukturentwürfe zu ermöglichen. Diese Effizienzsteigerung wird in der Regel durch den Einsatz von Regelungs- und Aktuatorsystemen erreicht, welche die Struktur gezielt durch das Einbringen zusätzlicher Belastungen oder Veränderung der Randbedingungen beeinflussen. Ziel dieser Arbeit ist die Kombination dieser beiden Themenfelder, um eine bestmögliche Regelbarkeit von Strukturen zu erreichen.

Zu diesem Zweck werden Mechanismen diskutiert, die eine möglichst effiziente Deformation dünner Schalenstrukturen gewährleisten. Diese werden durch den Einsatz der parameterfreien Formoptimierung erzeugt, wobei der Einfluss unterschiedlicher Optimierungskonfigurationen diskutiert wird. Außerdem wird ein erweitertes Konzept zur Design-Kontrolle vorgestellt, welches den zusätzlichen Herausforderungen der Optimierung einer aktuierten Struktur Rechnung trägt. Als Anwendungsbeispiel dient ein experimenteller intelligenter Tragflügel.

Im letzten Abschnitt werden die Einsatzmöglichkeiten der Strukturoptimierung zur Generierung bistabiler Strukturen betrachtet. Bisher werden im Bereich bistabiler Strukturen hauptsächlich Schalenstrukturen mit konstanter Krümmung sowie gleichmäßiger Materialisierung betrachtet.

Im Rahmen dieser Arbeit werden die Methoden der Freiform- und Materialoptimierung eingesetzt, um neuartige bistabile Schalenstrukturen zu generieren.

Acknowledgment

This dissertation was written from 2009 to 2015 during my time as a research assistant at the Chair of Structural Analysis (Lehrstuhl für Statik) at the Technische Universität München.

In the first instance, I want to thank Prof. Kai-Uwe Bletzinger for giving me the opportunity to work in his research group. I want to thank him for giving me the freedom to follow my own ideas in research, but at the same time for giving fruitful input and inspiration when ever needed. I also want to thank Roland Wüchner for a lot of productive discussions.

A special thanks goes to Prof. Axel Schumacher, Prof. Horst Baier and Prof. Fabian Duddeck for their interest in my work and their participation in the examination jury.

I also want to thank my co-workers at the institute for the good teamwork and the nice work environment. Explicitly I want to mention my office colleagues Michael Breitenberger, Michael Fischer and Matthias Firl.

My research was funded by the Deutsche Forschungsgemeinschaft (DFG). This funding is gratefully acknowledged.

Finally, I want to thank my family for their support during my studies and my PhD.

Munich, September 2015
Helmut Masching

Contents

CHAPTER 1

Introduction

1.1 Structural optimization in product development

Computer based analysis of structures plays an important role in nowadays process of product development. As product development cycles become shorter, the classical prototype construction is more and more replaced by numerical simulation. For the designing engineer, it is of advantage to have access to optimization software already in an early phase of product design, in order to get ideas about possible design improvements. In this context, usual applications focus on an optimal usage of material, where structures are usually optimized with respect to an optimal weight-to-strength ratio, or structural mass is reduced while keeping conditions with respect to maximum deflections or mechanical stresses.

1.2 Smart structures

1.2.1 Definition

Smart or adaptive structures are actively reacting mechanical systems, being able to respond to changes within their environmental conditions. By this mean, even more efficient and powerful structural designs are

enabled. Typically, smart structures are equipped with sensors, monitoring the state of the structure, and with actuators, that provide mechanical input and enable the structure to "react" to the actual environmental conditions [42]. The actuator reaction usually is linked to the sensor input data via a controller, which regulates the structure such that its overall behaviour, generated by external loading and the actuator influence, is optimal with respect to a pre-defined criteria [60].
Figure 1.1 shows the basic components of a smart structure. Senors, actuators and controller can be identified as central blocks, connected via data streams. These data streams transmit information gathered by the senors to the controller, and a corresponding actuator instruction is generated. Actuator input and sensor information are linked via the mechanical behaviour of the structure.

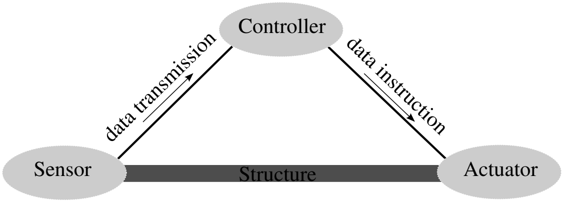


Figure 1.1: Basis components of a smart structure (according to [2])

1.2.2 State of the art

Smart structures are a current topic of intense research. The University of Stuttgart for example did a large-scale test of an adaptive light weight shell structure, originated from the field of civil engineering. The structure is a positively curved shell spanning over 10 meters, made of timber with a thickness of only 4 centimeters. The investigated structure is able to resist external loads, such as snow load, only due to specific actuating deformations at the supports [127].

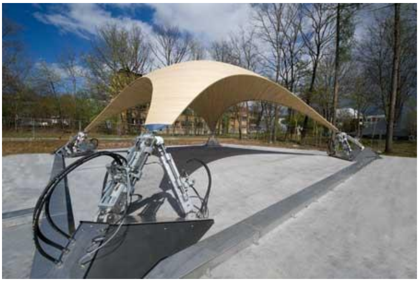


Figure 1.2: The adaptive "Smart Shell" structure in Stuttgart (source of picture: [127])

Another example of smart materials and structures applied in civil engineering is the Savannah River Site monitoring project [51], where highway bridges are monitored using piezoelectric or fiber optic sensors. In a second stage, the bridges are meant to be upgraded by using fiber reinforced plastic overlays.

Theoretical research regarding compliant shell mechanisms based on origami folding was done by Mark Schenk [109] and K. Seffen [113]. They considered multi-scale shell structures, consisting of repetitive meso cells such as corrugated bellows or "egg-box" segments, folded into sheet material. These meso cells undergo an almost strain free deformation, allowing easy deformability of the global structure.

Most applications of smart structures can be grouped into the field of aerospace applications. In these applications, mainly adaptive wings [23] or rotor blades [99] are considered. These airfoils are able to modify their shape such that they are optimally suited for the actual flight situation and flow condition of the surrounding air. One quite famous example for such an application in serial production is the flutter problem of the Boeing 747-8 aircraft. The new wide-bodied aircraft suffered from wing tip fluttering in certain unusual flight states, which endanger the aircraft's certification by the aviation regulator agency. The problem could be solved by a modification in the fly-by-wire system of the airplane, which detects the start of the flutter by acceleration sensors, and performs a slight movement of the ailerons in order to counter the vibrations of the flutter [62] [128]. Figure 1.3 illustrates the slender design of the wing and gives an impression about its sensitivity to vibrations.

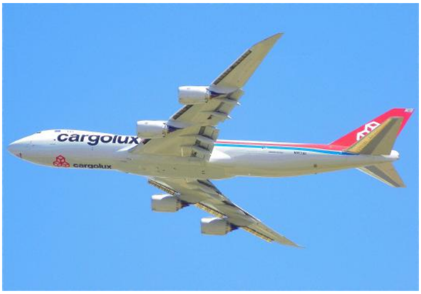


Figure 1.3: Highly flexible wing of the Boeing 747-8, prone to flutter (source of picture: [131])

Nowadays publications regarding optimization with respect to smart structures mainly focus on actuator placement for a given host structure using stochastic methods [130] or topology optimization [115]. As most applications of smart structures belong to the aerospace industry, a lot of publications regarding optimization do so as well. Farhan Gandhi and others are doing a lot of research work concerning morphing airfoils, where for example sizing optimization is applied to compliant mechanisms using honeycomb structures [61]. Bilgen and Friswell [22] [23] are working on a variable-camber wing for an unmanned light weight aircraft, where stochastic optimization algorithms are used on a parametrized model, but the focus of their work is on experiments.

1.3 Goal and outline of this thesis

This thesis aims on applying structural optimization to smart structures. The focus is on generating structures which provide an optimal response to the actuator input. Thereby, in contrast to other works mentioned in section 1.2.2, the entire structure is subject of optimization, and especially shape optimization of thin structures is a central aspect. For this reason, kinematic mechanisms of shell structures are discussed in more detail. In this context, several types of shells differing with respect to curvature are compared. Whereas Schenk [109] and Seffen [113] focused on a periodical meso-structure providing high flexibility, this thesis aims on generating localized compliant mechanisms in large scale structures.

The issue of actuator placement is included into the optimization process. Resulting effects onto the mathematical properties of the minimization problem are investigated, and a modified design control framework is presented.

In context of this work, the focus is put on the structure itself, not on the design of the controller which triggers the actuator input. Actuators are assumed to act in a statical manner, so that no time dependent controlling is considered. Information about optimal control and controller design can be found in Adamy [1] or Sontag [120], for example.

Another topic considered in the context of smart structures is the issue of bistability. A bistable structure is a structure which can assume different shapes of self equilibrium without any external load acting onto the structure. This property can be very interesting for smart structures, being able to switch between different configurations for different intends of use. Within this thesis, bistable structures are meant to be generated by using the parameter free optimization technique.

The further content of this thesis is divided into eight chapters:

Chapter 2 gives a very brief introduction into the fundamentals of the Finite Element Method. Thereby the focus is on introducing the element formulations used in this thesis.

Chapter 3 compares different mathematical methods used for optimization purpose. Pros and contras of different approaches are discussed, and an appropriate strategy for the intended purpose is chosen.

Chapter 4 is a mathematical chapter, giving a brief overview over solution techniques for constrained and unconstrained optimization problems using gradient based methods.

Chapter 5 introduces the parameter free optimization method, which is used in this thesis. The issue of shape and design control is addressed in detail, and the topic of an efficient and accurate sensitivity analysis is discussed. The chapter ends with a description of the data streams inside a modular object oriented analysis and optimization code framework.

Chapter 6 discusses optimal kinematic mechanisms for different types of structures, focusing on shells. For this purpose, different types of shells and different stress states are considered.

Chapter 7 addresses the issue of optimal actuation of structures. An extended design control framework is presented, allowing simultaneous shape optimization and actuator placement. The benefits of the parameter free approach are highlighted by generating complex bead designs, providing high flexibility in large deformation applications. The chapter ends with the consideration of a real world example.

Chapter 8 considers the topic of bistability. For this purpose, special characteristics of bistable structures and problems arising from the highly nonlinear load carrying behaviour are discussed. An appropriate response function is presented and derived, thereupon its fitness for the intended use is demonstrated for two curved shell structures.

Chapter 9 draws the final conclusions and gives a short outlook about possible future works.

1.4 Introduction to Carat++

All optimization and analysis results presented in this thesis are obtained by the institute's in-house finite element software Carat++ (**C**omputer **A**ided **R**esearch and **A**nalysis **T**ool). It is an object oriented research code written in C++, containing standard finite element analysis tools, beyond it includes modules for form finding and cutting patterning of membranes as well as a powerful optimization module. The origins of Carat go back to the late 1980s [84] [30], and a complete re-design in C++ was performed in 2008. Further information about Carat++ can be found in Masching et al [91] or Fischer et al [55].

CHAPTER 2

Finite Element Method in structural analysis

2.1 Structural analysis in the context of this work

In this work, analysis and evaluation of structural behaviour is an essential point. Any structural optimization approach would be useless without being able to evaluate the structure mechanical properties of the considered component. For this purpose, reliable and efficient numerical methods have to be applied, in order to obtain realistic simulation results of the mechanical behaviour of the investigated structure.

In the last decades, starting from the 1950s, a multiplicity of text books focusing on solving structure mechanical problems using numerical methods have been published. Primarily the Finite Element Method (FEM) was in focus of the authors. Giving a detailed insight about the fundamentals of these methods would go beyond the scope of this thesis and is also not necessary in this context. For these purposes, this chapter is meant to give a very brief introduction to state of the art numerical methods of structural analysis, but it is not claimed to be exhaustive. Detailed information about the Finite Element Method can be found, among others, in the textbooks by Zienkiewicz, Taylor [137] [138], Bathe [14], Argyris [6] [5] or Hughes [73].

2.2 Analytical equilibrium

The governing equation of structural analysis is the equilibrium condition. This equation is directly based on Newton's second law, which formulates the conservation of linear momentum. The equilibrium condition can be formulated as

$$div\left(\boldsymbol{\sigma}\left(\mathbf{u}\right)\right)+\rho\cdot\mathbf{b}=\rho\cdot\frac{d\mathbf{v}}{dt} \tag{2.1}$$

where $\boldsymbol{\sigma}\left(\mathbf{u}\right)$ is the stress state at the considered point of the structure, depending on its displacement $\mathbf{u}$. ρ defines the material density and $\mathbf{b}$ describes the body force vector. On the right hand side of the equation, $\mathbf{v}$ denotes the actual velocity of the point under consideration, so $\rho\cdot\frac{d\mathbf{v}}{dt}$ is the expression of the inertia forces.

Considering statical problems only, we can neglect the inertia term on the right hand side, and formulate the static equilibrium condition.

$$div\left(\boldsymbol{\sigma}\left(\mathbf{u}\right)\right)+\rho\cdot\mathbf{b}=0 \tag{2.2}$$

Equation 2.2 formulates the general equilibrium condition of a point inside an arbitrary continuum. This equation is called the *strong form of equilibrium*. In order to define the problem of structural static equilibrium properly, additional boundary conditions prescribing the deformation $\mathbf{u}$ (Dirichlet boundary conditions) or a derivative of the displacement field (Neumann boundary condition) at certain points have to be applied. These boundary conditions correspond to supports or local loads influencing the structure.

Unfortunately, equation 2.2 cannot be solved in the general case for arbitrary boundary conditions in order to determine the displacement $\mathbf{u}$ as an analytical function. In order to obtain an approximated solution for this problem nevertheless, numerical methods are applied.

2.3 The principle of Finite Element formulations

2.3.1 Weak form of equilibrium

A very successful approach to solve the equilibrium condition introduced in equation 2.2 is *Galerkin's Approach* [44] [49]. It is a numerical method solving operator equations, such as differential equations, approximately by applying a method of weighted residuals.

As Galerkin's Approach is an approximation method, equation 2.2 will not evaluate to zero at every point for the obtained approximated displacement field $\mathbf{u_h}$. The method weakens the requirements formulated in equation 2.2, such that it only demands equation 2.2 to be fulfilled in the integral mean over the complete domain Ω. An additional weighting of the residual is performed, using the so-called test function $\boldsymbol{\eta}$.

$$\int\limits_{\Omega} \left[div\left(\boldsymbol{\sigma}\left(\mathbf{u_h}\right)\right) + \rho \cdot \mathbf{b}\right] \cdot \boldsymbol{\eta} \; d\Omega = 0 \tag{2.3}$$

Using the variation of displacements $\delta\mathbf{u}$ as test function, the well known *weak form of equilibrium* is obtained:

$$\int\limits_{\Omega} \left[div\left(\boldsymbol{\sigma}\left(\mathbf{u_h}\right)\right) + \rho \cdot \mathbf{b}\right] \cdot \delta\mathbf{u} \; d\Omega = 0 \tag{2.4}$$

2.3.2 Discretization of geometry and displacement

In section 2.3.1, the equilibrium condition from equation 2.2 was transferred to the weak form, using an approximate solution $\mathbf{u_h}$ of the unknown displacement field $\mathbf{u}$. In order to be able to compute this approximate solution, it is necessary to reduce the continuous field $\mathbf{u}$ to a finite number of evaluation points, for which the approximate solution $\mathbf{u_h}$ is evaluated. As the weak form of equilibrium is formulated as an integral over Ω, a discretized description of the domain Ω is needed, too, in order to evaluate the integral equilibrium equation.

For the discretization of 2D-structures, for example an approximation of the surface using triangles or quadrilaterals is used, where the vertices of these facets are used as evaluation points for the displacement approximation as well as sampling points for the geometry description. These triangular or quadrangular facets are referred to as *finite elements* and the vertices are usually called *finite element nodes*. Figure 2.1 shows a discretization of a surface example using a coarse and a fine level of discretization. In this figure, also a template of a four-noded finite element is presented, showing the node numbering and a local coordinate system $\xi_1, \xi_2 \in [-1, 1]$, which allows to address any point inside the element by its local coordinates.

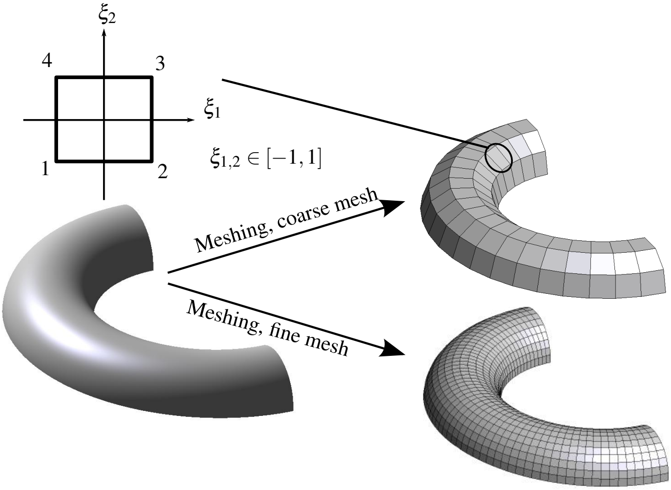


Figure 2.1: Discretization: analytical surface description, coarse and fine discretization using finite elements

Having this discretization information, geometry **X** or displacement **u** can be computed within an element by interpolating nodal information using interpolation functions, so-called *shape functions*,

$$\mathbf{u_h}\left(\xi_1,\xi_2\right) = \sum_{i=1}^{nNodes} N_i\left(\xi_1,\xi_2\right)\cdot\hat{\mathbf{u}}_\mathbf{i} \tag{2.5}$$

$$\mathbf{X_h}\left(\xi_1,\xi_2\right) = \sum_{i=1}^{nNodes} N_i\left(\xi_1,\xi_2\right)\cdot\hat{\mathbf{X}}_\mathbf{i} \tag{2.6}$$

where the index h emphasizes that an approximation is considered. N_i, $\hat{\mathbf{X}}_\mathbf{i}$ and $\hat{\mathbf{u}}_\mathbf{i}$ denote the shape function, the coordinates and the displacements belonging to node i, respectively.

2.4 Finite Element formulations used in this contribution

In this thesis, different kinds of structures are treated. Thin and light weight structures are considered as well as solid structures. For this reason, different types of finite element formulations are used.

2.4.1 Multi-layer Reissner-Mindlin shell element

For analysis and optimization of thin and wide span structures, a Reissner-Mindlin shell formulation is used, based on the thesis of Manfred Bischoff [24]. The element is a multi-layer degenerated solid shell, based on nonlinear, three-dimensional continuum theory. Thus, it is possible to use arbitrary three-dimensional constitutive laws without reduction or manipulation in nonlinear analysis of moderately thin structures including large deformations. In context of this thesis, linear elastic isotropic or orthotropic material laws using two or nine independent material parameters [79] are used.

The element uses a 7-parameter concept in order to be able to consider thickness changes as well as non-constant normal stress in thickness direction. Element nodes possess six degrees of freedom, namely three translations evaluated on the mid-plane of the shell structure, and three additional translatoric degrees of freedom describing the movement of the shell surface related to the mid-plane, thus the change of the shell director (see figure 2.2). The 7th degree of freedom describes linear normal strain distributions in thickness direction. This degree of freedom is condensed out on element level and does not appear in the element stiffness matrix explicitly. Linear ansatz spaces are used, where a triangular as well as a quadrilateral element configuration is available, and the the element formulation is equipped with ANS (assumed natural strains) [16] [15] and EAS (enhanced assumed strains) [116] improvements in order to get rid of locking phenomena and to increase result quality.

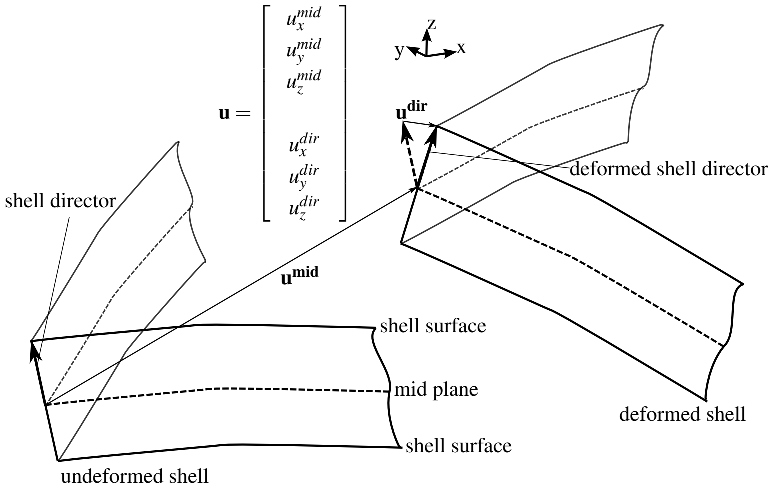


Figure 2.2: Definition of nodal degrees of freedom for Reissner-Mindlin shell element

Used as a composite element, the element acts as a single director multilayer shell, performing a layer-wise pre-integration of the material and stress tensors [32]. By this mean, the number of degrees of freedom is independent of the number of material layers used over the thickness of the element. As the integration is performed layer-wise, but the moment of inertia has to be captured correctly for each ply, different types of coordinate systems are used in shell thickness direction. The overall thickness coordinate $\theta^3 \in [-1,1]$ captures the complete shell thickness h, while for each ply i an additional coordinate $\theta_i^3 \in [-1,1]$ is defined, which spans the thickness h_i of the related ply. An example of a multi layer shell stack including the different coordinate systems is shown in figure 2.3.

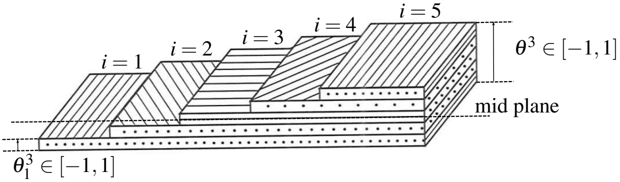


Figure 2.3: Multiple layer setup for composite element with layer and element coordinate systems (parts of figure taken from [21])

The integrated material tensor can be written in the form

$$D_k^{ijkl} = \sum_{i=1}^{n_{layers}} \int\limits_{\theta_i^3=-1}^{1} \left[\theta^3\left(\theta_i^3\right)\right]^k \cdot C^{ijkl} \cdot \mu \cdot \frac{h_i}{h}\ d\theta_i^3 \tag{2.7a}$$

$$\theta^3\left(\theta_i^3\right) = \frac{-h_i \cdot \left(1-\theta_i^3\right) + 2 \cdot \sum_{j=1}^{i} h_j}{h} - 1.0; \tag{2.7b}$$

$$\mu \approx \frac{h}{2} \tag{2.7c}$$

where equation 2.7b defines the mapping between the ply coordinate and the overall thickness coordinate. Equation 2.7c shows the shell shifter, which maps the dimension less coordinate θ^3 to the physical thickness of the shell.

Due to the single director description and the Reissner-Mindlin assumption (first-order shear deformation theory), the element cannot represent a warping of the cross section due to shear. These warping effects may arise especially for thick layered shell structures with distinct orthotropic material properties. In order to represent the shear deformation of the cross section more precisely, multi-layer shell models could be applied [102]. These models approximate the deformation of the cross section linearly in each ply, leading to a zick-zack shaped deformation over the entire thickness. Alternatively, higher order shear deformation theories can be used, describing the deformation in shell thickness direction using higher order polynomials [88] [108].

2.4.2 Three dimensional continuum element

Continuum elements are usually used in order to analyze structures which cannot be reduced to one or two dimensions, as their geometrical size is similar in all three directions in space. In this contribution, a hexahedral element formulation is used, including geometrical nonlinear kinematics. For the hexahedral element type, eight-noded elements are used applying trilinear shape functions. As these elements are known to suffer from geometrical locking effects due to parasitic strain [133] [125], the formulation is improved by the EAS (enhanced assumed strain) method as presented by Simo, Rifai [116] or Andelfinger, Ramm [3]. By this mean, the element is able to represent a trilinear stress states although the displacement interpolation itself is just linear. Therefore high computational accuracy can be achieved without increasing the number of element degrees of freedom.

CHAPTER 3

Mathematical optimization methods

3.1 Introduction to optimization

Optimization is a branch of mathematics which deals with the problem of finding the minimum or maximum value of a function $f(\mathbf{x})\,,\quad f: D \to \mathbb{R}$ and the related $\mathbf{x} \in \Omega$. Here, $\mathbf{x}$ is the vector of design variables, D denotes the design space of the optimization problem, and Ω is the set of feasible designs [11], thus a sub set of the design space. In literature about applied optimization, the function f is often referred to as *objective function* [67]. In context of this thesis, design variables are assumed to be real numbers, so the design space D is a higher dimension of $\mathbb{R}$.

$$D = \mathbb{R}^n \tag{3.1}$$

Accordingly, the feasible set is a sub set of $\mathbb{R}^n$

$$\Omega \subseteq \mathbb{R}^n \tag{3.2}$$

The set Ω can be defined either explicitly, or in an implicit way by adding additional equations or inequalities, so-called constraints, to the optimization problem, which a feasible design $\mathbf{x} \in \Omega$ has to fulfill. This will be discussed in more detail in section 4.3.

Most optimization procedures are designed such that they focus on minimization problems. Hence the standard optimization task can be formulated as the following minimization problem:

$$\begin{aligned} & f(\mathbf{x}), \quad f : \mathbb{R}^n \to \mathbb{R} \\ & min \ f(\mathbf{x}) \\ & with \ \ \mathbf{x} \in \Omega, \quad \Omega \subset \mathbb{R}^n \end{aligned} \tag{3.3}$$

In case the problem to solve is a maximization problem, it can easily be transformed into a minimization problem by applying a standard minimization algorithm to the negative function.

$$max \ f(\mathbf{x}) \Leftrightarrow min \ -f(\mathbf{x}) \tag{3.4}$$

As the transformation from maximization to minimization problems (and inverse) can be done in such an easy way, the following part of this chapter will only focus on the standard case of minimization problems.

3.2 Optimization methods

There are multiple ways to solve an optimization problem as it is defined in equation 3.3. The related methods are usually grouped according to the highest order of derivative which is used in the algorithm.

3.2.1 Zero order methods

Zero order methods are methods only evaluating the 0^{th} derivative of the objective function, which is the function value itself. The algorithms evaluate a multiplicity of candidate solutions and judge for each candidate solution if it is (a) feasible and (b) a minimum solution. A sub-grouping of zero order algorithms can be done according to the determination of the candidate solutions.

3.2.1.1 Direct search methods

Direct search methods just evaluate a specific set of given candidate solutions in order to determine the minimum feasible solution. The set of candidate solutions can either be generated by random (Monte-Carlo-Search) or by using a defined multi-dimensional grid of evaluation points (grid-search). Drawbacks of this very simple approach are the high number of response function evaluations and the restriction to discrete evaluation points, especially in a continuous design space.

3.2.1.2 Evolutionary Algorithms

The origin of genetic algorithms goes back to the 1950s, when researchers in biology started to simulate genetic processes using computers [59]. Soon the optimizing character of these processes became obvious. The basic principle of Evolutionary Algorithm can be compared to Darwin's evolutionary theory, which assumes a natural selection, leading to the survival and reproduction of these individuals which are best adapted to the ruling environmental conditions ("survival of the fittest").

An Evolutionary Algorithm starts from an arbitrary or randomly chosen initial population, which is a set of candidate solutions in the design space. For these candidate solutions, a fitness assignment according to the related objective function value is performed, and based on this assignment the fittest individuals are chosen for reproduction. Reproduction is based on two operations, crossover and mutation. Crossover is a pure re-combination procedure and produces a descendant based on the attributes of two parent individuals, while the mutation operator adds some statistical noise to the attributes of the descendant [95] [9]. After reproduction, all descendants are added to the population, and this is how the algorithm approaches the optimal solution over the generations of descendants.

By combining pure re-combination and mutation, the algorithm produces the best compromise of the initial population's attributes, but also the rest of the design space is investigated by the stochastic component of the mutation operator.

3.2.1.3 Particle Swarm Optimization

Particle swarm optimization is a relatively new method of optimization, which was presented in 1995 by Kennedy, Eberhart and Shi [82]. It is a method trying to mimic the behaviour of birds or fishes inside a swarm in order to detect the global optimum of an optimization problem [83].
Similar to genetic algorithms, particle swarm optimization is also based on a population, which is usually called "the swarm" in this context. Each candidate solution or particle of the swarm is moving through the design space following a velocity vector $\mathbf{v_i}$, which is computed as a linear combination of
(a) the connection vector from the actual particle position to the "best" position with respect to the objective function which the particle ever has reached and
(b) the connection vector from the actual particle position to the "best" position all particles of the swarm have discovered.
Summarizing, the idea of the algorithm is that each particle does not only act based on its local knowledge about the design space, but it also profits from information gained by other members of the swarm.

3.2.2 Methods of first order

In contrast to the methods mentioned in section 3.2.1, methods of 1^{st} order do not only evaluate the objective function itself, but also the information provided by the first derivative of the objective or constraint functions.
Compared to zero order function evaluations, which is a pure local information about the response function, information of 1^{st} order also gives information about the actual change of the response function when the design is varied.

Using first order information, directions of steepest ascend or steepest descend of the response function can be determined, as well as directions where the response function can be assumed not to change at all, as long as the investigated region is sufficiently close to the point of evaluation. Making use of this kind of information allows to tremendously reduce the number of objective function evaluations compared to zero order methods. The requirements for applying 1^{st} order methods is that the design variables of the optimization problem behave continuously within the design space, and that the objective function is C_1-continuous. When these conditions are fulfilled, the information of 1^{st} order can be computed for any design $\mathbf{x}$ in the design space.

3.2.3 Methods of higher order

In the previous section, methods of 1^{st} order were introduced, motivated by a gain of information, which was obtained by taking the first derivative of the objective function into account. A straight-forward extension of this approach leads to the development of methods of higher order, which do not only evaluate first derivatives information of the objective function, but also higher derivatives of order n.

Very famous methods of higher order are 2^{nd} order methods like Newton Methods, which also take into account the second order information of the response function. The second derivatives of the objective function f are usually collected in the so-called Hesse Matrix:

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_n} \\ \vdots & & & \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_n} \end{bmatrix} \tag{3.5}$$

Newton Methods will be addressed in more detail in chapter 4.

The requirements for applying higher order methods are similar to the requirements presented for the methods of 1^{st} order. Thus the design variables have to be continuous, and the objective function has to be sufficiently smooth. It has to satisfy at least C_n-continuity, in order to provide the required data over the complete design space.

3.3 Pros and contras of the different methods

Subsequently, pros and contras of the different approaches are compared. The benefits of methods of zero order are that they are very robust and may be combined with arbitrary black box software, as the required input information is reduced to the value of the objective function. This makes these methods also applicable to a non-continuous design space, where methods of first or higher order cannot be applied due to the requirements with respect to continuity mentioned in sections 3.2.2 and 3.2.3. Another benefit of zero order methods is that usually a population or a set of candidate solutions is used, which automatically leads to a broad exploration of the design space. This increases the likelihood of the method to converge to the global optimum of the problem.

On the other hand, methods of zero order require a high number of response function evaluations, especially when a large number of optimization variables is used. At this point, gradient based methods are in advantage towards zero order methods, as usage of information of first or higher order reduced the number of necessary objective function evaluations tremendously. At the same time, gradient based methods suffer from the previously mentioned drawbacks, that the objective function has to be smooth enough to provide the desired information, and, in addition, the computation of the gradient information requires detailed knowledge about the response function and access to the computational framework, which standard black box computer programs do not provide by default. An efficient computation of the required derivatives is a key feature to first or higher order methods. In sections 5.4 and 5.5, important aspects considering numerical efficiency of the computation of response function derivatives are discussed.

3.4 Conclusion and selection of optimization strategy

In the context of this thesis, an optimization approach based on a finite element analysis will be used, that supersedes the necessity of a separate model in order to define design parameters (detailed information will be given in chapter 5). While providing a maximum of freedom in design, this approach leads to a large number of optimization variables. Thereby the number of design parameters is equal to the number of finite elements or finite element nodes, so $10,000$ to $100,000$ design parameters are easily reached for usual applications. Following the considerations in section 3.3, a zero order optimization strategy would not be suitable for these kinds of problems, due to the large number of required objective function evaluations. The problems of consideration are sufficiently smooth, such that methods of first or second order can be applied. As computations are performed in a self-developed software framework providing full source-code access, derivatives of objective functions can be computed efficiently and with high accuracy. However, computation of higher order information based on a finite element model would be time consuming and numerically disputable, as already first order information requires special treatment, which will be discussed in proceeding chapters.

For these reasons, this work will focus on the usage of first order optimization strategies.

CHAPTER 4

Gradient based optimization

4.1 Classification

In context of this thesis, gradient based optimization is used as a collective term for all kind of optimization algorithms where the first or second order derivatives are taken into account.

4.2 Unconstrained optimization

4.2.1 Optimality criteria

Unconstrained optimization is the most simple case of an optimization problem. In this case, there are no restrictions defined which shrink the feasible set. Consequently the feasible set is equal to the design space.

$$\Omega = D \tag{4.1}$$

Furthermore, we now assume that f is C_1-continuous in D. Based on this assumption, the necessary condition for f being minimal in $\mathbf{x}_0$ is that all first partial derivatives at this point arise to be zero.

$$\frac{\partial f}{\partial x_i} = 0 \;\; \forall \;\; i \in \{1, ..., n\} \tag{4.2}$$

Usually the first partial derivatives of a function are collected in the gradient vector. Usage of the gradient vector allows to reformulated the optimality condition as

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f}{\partial x_1}\ \frac{\partial f}{\partial x_2}\ \dots\ \frac{\partial f}{\partial x_n}\right]^T = \mathbf{0} \tag{4.3}$$

A sufficient condition can be formulated for C_2-continuous functions, by adding the additional requirement for the Hesse Matrix to be positive definite.

4.2.2 Solution methods

4.2.2.1 Newton Methods

"The Newton Method is a cornerstone of numerics and it can be used for solving nonlinear systems of equations as well as for minimizing nonlinear functions." (from Ulbrich [126], translated by the author) In order to determine the unconstrained optimum, the nonlinear system of equations presented in equation 4.3 has to be solved. The Newton Method is a classical and very successful method to solve this kind of problem. It performs an approximation of the original problem using a linear Taylor series. Solving this linear problem provides an improved approximate solution of the initial problem. By solving these of linearized problems iteratively, the Newton Method approaches the solution of the initial problem [48]. Applied to equation 4.3, the corresponding iteration scheme reads

$$\nabla f\left(\mathbf{x}^i\right) + \underbrace{\nabla\left(\nabla f\left(\mathbf{x}^i\right)\right)}_{\mathbf{H}\left(\mathbf{x}^i\right)} \cdot \Delta\mathbf{x} = \mathbf{0} \tag{4.4a}$$

$$\mathbf{x}^{i+1} = \mathbf{x}^i + \Delta\mathbf{x} \tag{4.4b}$$

where i denotes the current iteration, and $\mathbf{x}^i$ is the approximated solution vector in the i^{th} iteration. The approximate solution for the evaluations for the next iteration step is computed according to equation 4.4b.

Linearization of equation 4.3 leads to second derivatives of the objective function f, collected in the Hesse Matrix $\mathbf{H}$, hence Newton Methods belng to *optimization strategies of second order*.
The necessity to compute the second order information is a drawback of the Newton Method. In cases where, for example, the function f is not known in an analytical form but only can be evaluated numerically, the computation of second order derivatives may become very time consuming or even impossible. In order to overcome this drawback, so-called Quasi-Newton-Methods have been developed, which perform an approximation of the Hessian matrix or even the inverse Hessian matrix, as in fact the inverse matrix is needed in order to compute the incremental vector $\Delta\mathbf{x}$. One very popular Quasi-Newton-Method is the so-called BFGS-Method, which was developed in 1970 by the mathematicians Broyden [33], Fletcher [57], Goldfarb [64] and Shanno [114] independently from each other.

4.2.2.2 Descent methods

Another method to overcome the problems of the classical Newton Method is to completely neglect the second order information and apply pure *descent methods* or *algorithms of first order*, which only use gradient information only in order to determine the function minimum. For this purpose, we do not focus on the multi-dimensional optimality criteria from equation 4.3 directly, but the optimization procedure is considered a sequence of one-dimensional optimization problems with respect to a chosen search direction. Metaphorically speaking, the algorithm starts at a given point $\mathbf{x}$ in D and "moves" through the design space along a chosen search direction vector $\mathbf{s}$ as long as the function value decreases in direction $\mathbf{s}$. This one-dimensional optimization problem (also called *line search*) determines a step-length parameter α such that the total derivative of f at point $\mathbf{x} + \alpha \cdot \mathbf{s}$ in direction of $\mathbf{s}$ vanishes:

$$[\nabla f(\mathbf{x}+\alpha\cdot\mathbf{s})]^T\cdot\mathbf{s}=0 \tag{4.5}$$

Assuming $\mathbf{s}$ to be a descent direction such that $\left[\mathbf{s}^{\mathbf{i}}\right]^T\cdot\nabla f\left(\mathbf{x}^{\mathbf{i}}\right)<0$ [126], the step length determined in the line search has to be strictly positive.

$$\alpha>0 \tag{4.6}$$

The optimal point found for the current search direction $\mathbf{s}^i$ is used as starting point for the next optimization iteration $i+1$,

$$\mathbf{x}^{i+1}=\mathbf{x}^i+\alpha\cdot\mathbf{s}^i \tag{4.7}$$

and the procedure is repeated until the optimum point is reached. Figure 4.1 shows a corresponding pseudo code.

```
choose a starting point $\mathbf{x}^0$;
$i=0$;
converged = false;
while not converged do
    choose descent direction s at point $\mathbf{x}^i$;
    1D minimization along s: $\left[\nabla f\left(\mathbf{x}^i+\alpha\cdot\mathbf{s}\right)\right]^T\cdot\mathbf{s}=0$ (solve for $\alpha>0$) ;
    Update x for next iteration: $\mathbf{x}^{i+1}=\mathbf{x}^i+\alpha\cdot\mathbf{s}$;
    if $\nabla f\left(\mathbf{x}^{i+1}\right)=\mathbf{0}$ then
        converged = true;
    end
    $i=i+1$;
end
```

Figure 4.1: Optimization algorithm for a general descend method

It is obvious that a good choice of the search direction is the key issue of this procedure. An arbitrary chosen descent direction already leads to a decrease of the objective function, but of course a more sophisticated choice of the search direction has to be made. In order to obtain the most rapid decrease of f along the search direction, it is appropriate to choose a search direction which is parallel to the gradient vector. This is done in the *steepest descent method*, where the search direction is chosen as the negative gradient [100].

$$\mathbf{s}^i = -\nabla f\left(\mathbf{x}^i\right) \tag{4.8}$$

A characteristic property of steepest descent is that search directions of following iteration steps are orthogonal to each other,

$$\left(\mathbf{s}^{i+1}\right)^T \cdot \mathbf{s}^i = 0 \tag{4.9}$$

as equation 4.5 holds and the new search direction will be chosen as $\mathbf{s}^{i+1} = -\nabla f\left(\mathbf{x}^i + \alpha^i \cdot \mathbf{s}^i\right)$. As a consequence, steepest descent shows a very good convergence behaviour as long as the eigen values of the Hessian are identical. Thus, "optimality" which was reached with respect to one search direction is not destroyed by any further optimization step [8]. Otherwise interaction of search directions slows down the convergence rate of the steepest descent algorithm. This can be seen by looking at the orthogonality of search directions with respect to the Hessian matrix of f,

$$\left(\mathbf{s}^i\right)^T \cdot \mathbf{H} \cdot \mathbf{s}^j \neq 0, \quad i \neq j \tag{4.10}$$

which is not fulfilled in general for steepest descent search directions.
In order to maintain a high rate of convergence also for objective functions with badly conditioned Hesse Matrices, Fletcher and Reeves [58] published a method of *conjugated gradients (CG)*, which takes previous search directions into account in order to compute a new one. By this mean, search directions are not orthogonal to each other anymore, but satisfy equation 4.10. Hence it can be proven that the CG method converges to the minimum in n steps for quadratic objective functions $f : \mathbb{R}^n \to \mathbb{R}$. [8]

Equation 4.11 shows the computation of the i^{th} search direction depending on the previous one according to Fletcher and Reeves [58].

$$\begin{aligned} \mathbf{s}^i &= -\nabla f\left(\mathbf{x}^i\right) + \beta \cdot \mathbf{s}^{i-1} \\ \beta &= \frac{\nabla f\left(\mathbf{x}^i\right)^T \cdot \nabla f\left(\mathbf{x}^i\right)}{\nabla f\left(\mathbf{x}^{i-1}\right)^T \cdot \nabla f\left(\mathbf{x}^{i-1}\right)} \end{aligned} \tag{4.11}$$

In figure 4.2, the convergence behaviour of different descent methods is compared for the function $f(x,y) = x^2 + 4 \cdot y^2$. The left picture shows that even a method using a randomly chosen search direction satisfying $\left[\mathbf{s^i}\right]^T \cdot \nabla f\left(\mathbf{x^i}\right) < 0$ converges to the optimum, as the 1D optimization procedure reduces the objective value in every single step, but the speed of convergence is very slow and the convergence behaviour appears arbitrary. The steepest descent method shows the "zick-zacking" which typically appears for problems with a badly conditioned Hesse Matrix, and it can be seen that search directions are orthogonal to each other. In contrast, for the CG method the search directions are not orthogonal, but H-orthogonal according to equation 4.10. As a consequence, the procedure converges as expected in the second step as a two-dimensional quadratic problem is considered.

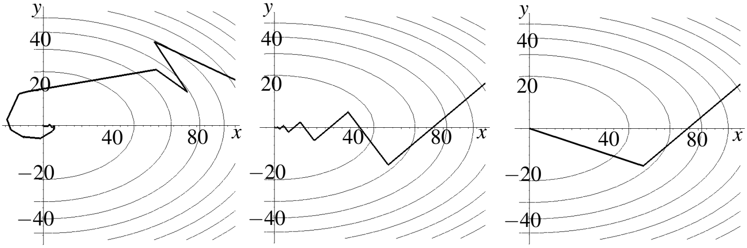


Figure 4.2: Comparison of descent methods. Left: Randomly chosen search directions, Center: Steepest Descent, Right: Conjugate Gradient

4.3 Constrained optimization

4.3.1 Definition of constraints

The concept of constraint optimization was already briefly discussed in section 3.1, when the idea of the feasible set was introduced. The feasible set was defined in a very abstract way, as a sub set of the design space, $\Omega \subseteq D$. In applied optimization, the definition of the feasible set is usually done by adding additional equations or inequalities to the optimization problem, so-called equality or inequality constraint functions, which a $\mathbf{x} \in \Omega$ has to fulfill. Using the vectors of constraint functions $\mathbf{g}$ and $\mathbf{h}$, the abstract formulation of the constrained optimization problem (equation 3.3) can be written in the following form:

$$\begin{aligned} &min\ f(\mathbf{x}) \\ &st \\ &\mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ &\mathbf{h}(\mathbf{x}) = \mathbf{0} \end{aligned} \tag{4.12}$$

4.3.2 Lagrange function

The idea of the Lagrange function is to transfer a constrained optimization into a unconstrained problem and to determine a gradient based optimality criteria for the general constrained optimization problem, similar to equation 4.3. In order to explain the idea of the Lagrangian formulation, we consider the following two-dimensional optimization problem of a quadratic function under consideration of a linear equality constraint (illustrated in figure 4.3):

$$\begin{aligned} &min\ f;\ \ f : \left(\mathbb{R}^2 \rightarrow \mathbb{R}\right) \\ &st \\ &h : a \cdot x + b - y = 0 \end{aligned} \tag{4.13}$$

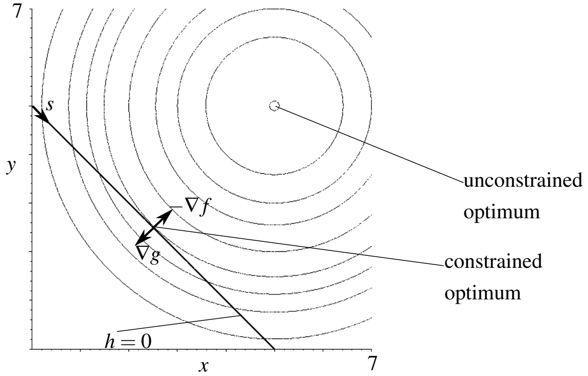


Figure 4.3: Example of constrained optimization

In order to solve this problem, a one dimensional optimization along the zero contour line of the constraint function (which is given by the vector **s** in figure 4.3) is performed, as it was described in section 4.2. For this problem, the one dimensional optimization will lead directly to the constrained optimum, as the feasible set is given by the linear contour line $h = 0$. Thus the following condition can be formulated for the optimum of this constrained problem,

$$\begin{bmatrix} \mathbf{s}^T \cdot \nabla f \\ h \end{bmatrix} = \mathbf{0} \tag{4.14}$$

whereat the first line specifies the optimality of the point under investigation, and the second line guarantees its feasibility.

As the contour line direction **s** is normal to the gradient of the constraint ∇h, the orthogonality condition in equation 4.14 can be replaced by the condition that ∇h and ∇f have to be collinear at the constrained optimum, where μ is the collinearity factor:

$$\begin{bmatrix} \mu \cdot \frac{\partial h}{\partial x} + \frac{\partial f}{\partial x} \\ \mu \cdot \frac{\partial h}{\partial y} + \frac{\partial f}{\partial y} \\ h \end{bmatrix} = \mathbf{0} \tag{4.15}$$

A generalized derivation of this optimality condition, based on the idea of using tangential cones in order to describe the feasible domain, can be found in Ulbrich [126].

Computing the potential of equation 4.15 leads to the so-called Lagrange function $L(x,y,\mu) = f + \mu \cdot h$, and the optimality criteria from equation 4.15 can be formulated via the gradient of L with respect to x, y and μ:

$$\begin{bmatrix} \mu \cdot \frac{\partial h}{\partial x} + \frac{\partial f}{\partial x} \\ \mu \cdot \frac{\partial h}{\partial y} + \frac{\partial f}{\partial y} \\ h \end{bmatrix} = \nabla_{\mathbf{x},\mu} L = \mathbf{0} \tag{4.16}$$

Applying this approach to a general constrained optimization problem with arbitrary numbers of equality and inequality constraints, the Lagrange function can be formulated in the following form [101]:

$$L(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\tau}) = f(\mathbf{x}) + \boldsymbol{\mu} \cdot \mathbf{h}(\mathbf{x}) + \sum_i \lambda_i \cdot \left(g_i(\mathbf{x}) + \tau_i^2\right) \tag{4.17}$$

τ_i^2 represents the so-called slack variables, which allow to treat inequality constraints similar to equality constraints.

The values $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$ represent the so-called Lagrange multipliers, which represent the ratio between the gradients of constraints and the objective function at the optimum point, as it can be seen in equation 4.15 for the simple example. Alternatively, the design variables x are also called *primal variables*, whereas the multipliers $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$ are referred to as *dual variables*.

Computing the second derivatives of the Lagrange function, we observe that $\frac{\partial^2 L}{\partial \boldsymbol{\mu}^2} = 0$ and $\frac{\partial^2 L}{\partial \boldsymbol{\lambda}^2} = 0$. Accordingly the Hessian matrix of the Lagrange function will be indefinite, which means that the Lagrangian forms a saddle surface in the space of primal and dual variables.

The general first order necessary optimality criteria (FONC) [11] of the constrained optimization problem now can be formulated, according to equation 4.16, by the requirement of all first partial derivatives of the Lagrange function to evaluate to zero (equation 4.18).

$$\nabla_{\mathbf{x},\boldsymbol{\mu},\boldsymbol{\lambda},\boldsymbol{\tau}} L = \begin{bmatrix} \frac{\partial L}{\partial x_1} \\ \vdots \\ \frac{\partial L}{\partial x_n} \\ \frac{\partial L}{\partial \mu_1} \\ \vdots \\ \frac{\partial L}{\partial \mu_k} \\ \frac{\partial L}{\partial \lambda_1} \\ \vdots \\ \frac{\partial L}{\partial \lambda_l} \\ \frac{\partial L}{\partial \tau 1} \\ \vdots \\ \frac{\partial L}{\partial \tau l} \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x_1} + \sum_{j=1}^{k} \frac{\partial h_j}{\partial x_1} \cdot \mu_j + \sum_{j=1}^{l} \frac{\partial g_j}{\partial x_1} \cdot \lambda_j \\ \vdots \\ \frac{\partial f}{\partial x_n} + \sum_{j=1}^{k} \frac{\partial h_j}{\partial x_n} \cdot \mu_j + \sum_{j=1}^{l} \frac{\partial g_j}{\partial x_n} \cdot \lambda_j \\ h_1 \\ \vdots \\ h_k \\ g_1 + \tau_1^2 \\ \vdots \\ g_l + \tau_l^2 \\ 2 \cdot \lambda_1 \cdot \tau_1 \\ \vdots \\ 2 \cdot \lambda_l \cdot \tau_l \end{bmatrix} = \mathbf{0} \tag{4.18}$$

4.3.3 Interpretation of Lagrange multipliers

At the beginning of section 4.3.2, we considered an optimization problem which was constrained by one equality constraint. In that case, the constrained optimum was defined by the gradient of the objective ∇f being collinear to the gradient of the constraint ∇h, and the Lagrange multiplier μ turned out to be the collinearity factor of these two vectors. Now, the interaction of several constraints and the special role of inequality constraints will be addressed.

Equation 4.18 consists of two parts. The first part, which contains the derivatives with respect to the primal variables, enforces a point to be optimal, while the derivatives with respect to the dual variables in the second part enforce the constraints to be active. Assuming that all constraints are active and just considering the first part of equation 4.18, the negative gradient of the response function can be displayed as a linear combination of the gradients of the active constraints, where the Lagrange multipliers form the linear combination factors. This can be interpreted as a basis change, where $-\nabla f$ is transferred to the basis defined by the gradients of active constraints $\{\nabla g_1, \nabla g_2, ..., \nabla g_n\}$, and the Lagrange multipliers form the new components of $-\nabla f$.

$$-\nabla f = \sum_{j=1}^{n} \nabla g_j \cdot \lambda_j \tag{4.19}$$

Figure 4.4 illustrates this situation, using a linear objective function and two linear inequality constraints in a two dimensional design space. A graphical examination shows that both Lagrange multipliers turn out to be positive when expressing $-\nabla f$ with respect to the basis $\{\nabla g_1, \nabla g_2\}$

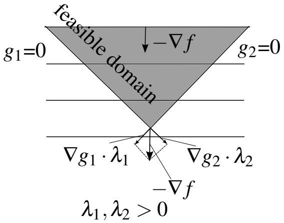

feasible domain
$-\nabla f$
g_1=0
g_2=0
$\nabla g_1 \cdot \lambda_1$
$\nabla g_2 \cdot \lambda_2$
$-\nabla f$
$\lambda_1, \lambda_2 > 0$

Figure 4.4: Positive Lagrange multipliers: Constrained optimum defined

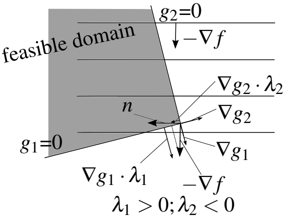

g_2=0
feasible domain
$-\nabla f$
$\nabla g_2 \cdot \lambda_2$
n
∇g_2
g_1=0
∇g_1
$\nabla g_1 \cdot \lambda_1$
$-\nabla f$
$\lambda_1 > 0; \lambda_2 < 0$

Figure 4.5: One Lagrange multiplier negative: No optimum

and the displayed situation represents a constrained optimum. In contrast, figure 4.5 shows a similar situation where equation 4.19 leads to a positive and a negative Lagrange multiplier, $\lambda_1 > 0, \lambda_2 < 0$. As a consequence, the vector $\mathbf{n} = \lambda_2 \cdot \nabla g_1 + (-\lambda_1) \cdot \nabla g_2$, which is normal to ∇f, [1] only consists of negative coefficients with respect to the basis $\{\nabla g_1, \nabla g_2\}$. Accordingly any movement in direction of $\mathbf{n}$ decreases both equality constraints, and both constraints become in-active, while the value of f remains constant. This makes the investigated point a non-optimal point.

[1] assuming $\{\nabla g_1, \nabla g_2\}$ form an orthonormal basis

In summary, these examples show that Lagrange multipliers associated to inequality constraints always have to be non-negative at the optimum point [96]. For equality constraints, of course, this condition does not hold, because the constraint equation can be arbitrarily scaled without influencing the feasible set or the optimum solution.

4.3.4 Karush-Kuhn-Tucker conditions

The awareness gained in the last section is combined with the optimality criteria shown in equation 4.18 now. This equation does not yet consider any additional conditions regarding inequality constraints, meaning all constraints are assumed to be active at the optimum. As an equality constraint can be inactive at the optimum and so it does not effect the solution, equation 4.18 has to be modified in order to get a general gradient based optimality criteria. For this purpose, inequality constraints are separated into two groups:

- *inactive inequality constraint:* $g_i < 0, \lambda_i = 0$
- *active inequality constraint:* $g_i = 0, \lambda_i > 0$

As a consequence, it is ensured that the product of inequality constraint value times Lagrange multiplier is always equal to zero, no matter if the constraint is active or not. Using this property, equation 4.18 can be reformulated to the following form:

$$\begin{aligned}
&\frac{\partial L}{\partial \mathbf{x}} = \mathbf{0} \\
&h_j = 0 \\
&g_i \cdot \lambda_i = 0 \text{ with } \lambda_i \geq 0,\ g_i \leq 0
\end{aligned} \tag{4.20}$$

This equation is called *Karush-Kuhn-Tucker-condition (KKT)*, named after the mathematicians Harold Kuhn and Albert Tucker, who published

the idea of the necessary condition in 1951 [85]. Later it turned out that William Karush already worked in this field in 1939 [80].
The difference to the previous version in equation 4.18 is, that now inactive inequality constraints can be considered in the formulation, too, and that the non-negativity of Lagrange multipliers associated to inequality constraints is part of the formulation.

4.3.5 Solution methods

There are several approaches to solve the constrained optimization problem. A selection of commonly used approaches is briefly presented in this section.

4.3.5.1 Newton-Lagrange Methods and Sequential Quadratic Programming (SQP)

The Newton-Lagrange Method is a quite obvious approach to solve the constrained optimization problem. Similar to the Newton Method applied to unconstrained problems (see section 4.2.2.1), it solves the nonlinear equation representing the optimality condition by applying the Newton Method. As the straight forward application of the Newton Method requires knowledge about the set of active constraints at the optimum, the Newton-Lagrange Method in its basic form usually is applied to equality constrained problems [126]. The corresponding linearization reads

$$\begin{bmatrix} \frac{\partial L\left(\mathbf{x}^i,\boldsymbol{\mu}^i,\boldsymbol{\lambda}^i\right)}{\partial \mathbf{x}} \\ \mathbf{h}\left(\mathbf{x}^i\right) \end{bmatrix} + \begin{bmatrix} \frac{\partial^2 L\left(\mathbf{x}^i,\boldsymbol{\mu}^i,\boldsymbol{\lambda}^i\right)}{\partial \mathbf{x}^2} & \frac{\partial \mathbf{h}\left(\mathbf{x}^i\right)}{\partial \mathbf{x}} \\ \left[\frac{\partial \mathbf{h}\left(\mathbf{x}^i\right)}{\partial \mathbf{x}}\right]^T & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\boldsymbol{\mu} \end{bmatrix} = \mathbf{0} \quad (4.21)$$

where i again is the iteration counter, and design as well as the Lagrange multipliers are updated similar to the Newton Method by $\mathbf{x}^{i+1} = \mathbf{x}^i + \Delta\mathbf{x}$ or $\mu^{i+1} = \mu^i + \Delta\mu$.

In case inequality constraints are involved and it is not known which constraints will become active, usually an auxiliary sequence of quadratic optimization problems is generated (Sequential Quadratic Programming). For this purpose, response and constraint functions are evaluated at a specific point, and a quadratic approximation of the response function and linear approximations of the active constraints ("active set") are performed. This auxiliary quadratic problem

$$\begin{aligned} &\underset{\mathbf{d}}{min} \quad f\left(\mathbf{x}^i\right)+\mathbf{d}^T\cdot\frac{\partial f\left(\mathbf{x}^i\right)}{\partial\mathbf{x}}+\frac{1}{2}\cdot\mathbf{d}^T\cdot\frac{\partial L\left(\mathbf{x}^i,\boldsymbol{\mu}^i,\boldsymbol{\lambda}^i\right)}{\partial\mathbf{x}}\cdot\mathbf{d} \\ &st \quad \mathbf{h}\left(\mathbf{x}^i\right)+\mathbf{d}^T\cdot\frac{\partial\mathbf{h}\left(\mathbf{x}^i\right)}{\partial\mathbf{x}}=\mathbf{0};\quad \mathbf{g}\left(\mathbf{x}^i\right)+\mathbf{d}^T\cdot\frac{\partial\mathbf{g}\left(\mathbf{x}^i\right)}{\partial\mathbf{x}}\leq\mathbf{0}\end{aligned} \tag{4.22}$$

is minimized iteratively, and a new active set is chosen for the next quadratic approximation [31].

4.3.5.2 Penalty methods

Penalty methods solve the constrained optimization problem without consideration of a Lagrange function or Karush-Kuhn-Tucker conditions. Instead, it penalizes violations of constraints. By this mean, an auxiliary objective function f_{pen} is generated, which can be solved as an unconstrained problem. A frequently used penalization method is the so-called quadratic exterior penalization [10], which adds quadratic penalization terms for violated constraints to the objective function

$$f_{pen}=f+\rho\cdot\sum_i h_i^2+\rho\cdot\sum_i max\left\{\begin{array}{c}0\\ g_i\end{array}\right\}^2 \tag{4.23}$$

Here, ρ is a penalty factor, weighting the added penalization terms. A benefit of this method is that it is very robust, but the drawback is that the constrained optimum only can be found theoretically by using an infinite large penalty factor. Consequently, the approximated solution will always be located outside the feasible domain for numerically reasonable penalty factors. Furthermore, the unconstrained minimization problem $min\ f_{pen}$ tends to be ill-conditioned for large penalty factors [11].

4.3.5.3 Dual methods

As it was already described in section 4.3.2, the Lagrange function describes a saddle surface in the space of primal and dual variables, where the solution of the constrained problem according to equation 4.18 is given by the saddle point of this surface. Due to this property, the constrained optimum cannot be found by applying a simple descent method like steepest descent or CG to the Lagrange function, as the solution is defined by a stationary point $\nabla L = \mathbf{0}$, which is not a local minimum.

The idea of dual methods is to determine the saddle point using a staggered optimization scheme which

- minimizes the Lagrange function with respect to the primal variables $\mathbf{x}$ (*inner problem*) and
- maximizes the Lagrange function with respect to the dual variables $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$ (*outer problem*) [11]

and approaches the saddle point iteratively.

The benefit of the dual approach is that, in contrast to the penalty approach, the optimum solution can be determined exactly, as it determines the Karush-Kuhn-Tucker point. The drawback is that convergence of the staggered optimization approach cannot be guaranteed in general.

4.3.5.4 Augmented Lagrangian Method (ALM)

The Augmented Lagrangian Method combines the idea of the dual method with a penalty approach in order to overcome the drawbacks of both methods, and so to formulate an optimization method which is guaranteed to converge and able to find the exact optimum for a finite penalty factor [96]. The ALM was discussed the first time in 1969 by Hestenes [71] and Powell [106].

The Augmented Lagrangian objective function L_{ALM} can be formulated in the following way

$$\begin{aligned} L_{ALM} &= f + \sum_i \left[h_i \cdot \mu_i + \frac{1}{2} \cdot \rho \cdot h_i^2 \right] + \\ &+ \sum_i \left\{ \begin{array}{cc} g_i \cdot \lambda_i + \frac{1}{2} \cdot \rho \cdot g_i^2, & g_i + \frac{\lambda_i}{\rho} \geq 0 \\ -\frac{1}{2 \cdot \rho} \cdot \lambda_i^2, & g_i + \frac{\lambda_i}{\rho} < 0 \end{array} \right\} \end{aligned} \tag{4.24}$$

where the terms related to the dual variable $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are inherited from the dual approach, and the terms multiplied with the penalty factor ρ show the components coming from the penalty approach.

Optimization again is done in a staggered way, as it was described for the dual method. In each iteration step k a minimization with respect to $\mathbf{x}$ is performed

$$\overset{min}{\mathbf{x}} \; L_{ALM} \left(\mathbf{x}, \boldsymbol{\mu}^k, \boldsymbol{\lambda}^k \right) \tag{4.25}$$

and afterwards the dual variables are updated

$$\begin{aligned} \mu_i^{k+1} &= \mu_i^k + \rho \cdot h_i(\mathbf{x}) \\ \lambda_i^{k+1} &= \lambda_i^k + \rho \cdot max \left\{ \begin{array}{c} 0 \\ g_i(\mathbf{x}) \end{array} \right\} \end{aligned} \tag{4.26}$$

representing the outer maximization problem with respect to the dual variables.

4.4 Applied algorithms

The numerical examples presented in this thesis are solved using either a *conjugated gradient* or an *augmented lagrangian* algorithm, depending on if a unconstrained or constrained optimization problem is considered.

CHAPTER 5

Concept of parameter free Shape Optimization

5.1 Branches of Structural Optimization

Structural Optimization is a very broad field with a multiplicity of possible applications. Sub-disciplines inside Structural Optimization can be defined and identified by distinguishing between different types of design parameters (see figure 5.1). In order to be able to delimit these disciplines

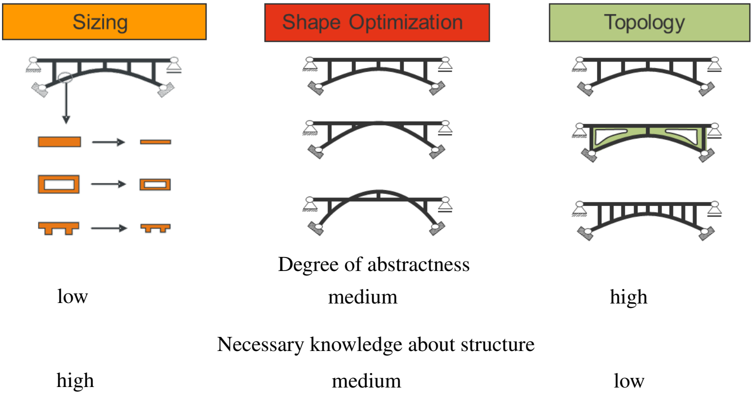


Figure 5.1: Different branches of Structural Optimization

against each other, two terms have to be defined which are used in order to describe a structure:
The *topology* of a structure is defined the type and connectivity of the structural members the structure consists of, such as trusses, beams, or plate components. For discrete or discretized structures, these members are connected at nodes.
The *shape* of a discrete or discretized structure is defined by the position of its nodes and the underlying topology.
Using these terms, the optimization disciplines illustrated in figure 5.1 easily can be delimited:

In *Sizing Optimization*, the shape of a structure and also the underlying topology is determined and remains unchanged. Only non-shape determining parameters are in focus of optimization. These parameters are usually cross section parameters like wall or plate thicknesses, pipe diameter or beam heights.
In contrast, *Topology Optimization* does not implicate any restricting boundary conditions with respect to shape or topology of the structure. The optimization problem is defined by a design space and boundary conditions, and the optimizer is completely free in finding a topology and a shape being optimal with respect to the chosen objective function for the given boundary conditions.
Shape Optimization works on a topologically fixed structure, whose shape is modified by moving the intersecting nodes of the structural members. Shape optimization includes sizing optimization.

This thesis focuses on shape optimization and its challenges of controlling shape and sizing parameters in a finite element attributed parametrization scheme.

5.2 Shape modification and design parametrization[1]

For the numerical treatment of problems in shape optimization, adequate parametrization of the design space is necessary. A meanwhile classical way of parameterizing design is to apply an extra CAD-model as a design model [97] [86]. This is a self-evident parametrization approach, as the CAD-model used for standard product engineering can be used in order to derive the parametrized optimization CAD-model [68] [112]. The idea of using NURBS or Bezier functions based shape descriptions in order to parametrize an optimization model is state of the art [28] [136]. Another classical parametrization approach is the usage of morphing boxes. Morphing is a technology which is originated in image manipulation. Thereby pixel clusters representing an images are warped using smooth thin plate splines (TPS) of minimum bending [117], whereat the deformation is controlled via a set of selected control points [37]. In the optimization context, this spline-based warping is used in order to control the shape of a finite element model instead of an image. So, design modifications can be regulated via the movement of the chosen control points [68].

In contrast, this thesis uses a node based parametrization approach. This approach defines optimization parameters directly on the finite element mesh which is used for structural analysis. A separate model for design parametrization is not necessary. Therefore, the name "parameter free" in the sense of "free of additional design parameters" became conventional. Now, the parameters for shape design are the coordinates of each finite element node or vertex in the context of this thesis [56] [72].

This finite element attributed parametrization concept is not reduced to shape determining parameters only. In principle, all parameters which can be identified in a finite element model can be used as design parameters directly, whereby this parametrization approach is also used for sizing optimization in this thesis. In the following explanations of this chapter,

[1] Parts of this section have been pre-published in Masching, Bletzinger [90]

the theory behind the parameter free optimization concept will be discussed focusing on shape determining parameters, as they bring up the major challenges with respect to design control.

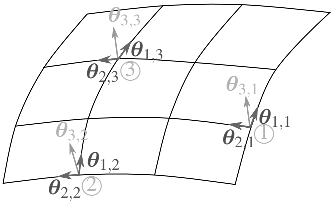
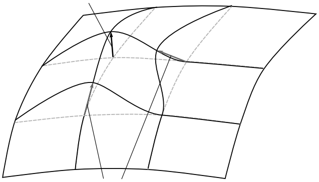

$\theta_{3,3}$
$\boldsymbol{\theta}_{1,3}$
$\boldsymbol{\theta}_{2,3}$
③
$\theta_{3,1}$
$\boldsymbol{\theta}_{1,1}$
$\boldsymbol{\theta}_{2,1}$
①
$\theta_{3,2}$
$\boldsymbol{\theta}_{1,2}$
$\boldsymbol{\theta}_{2,2}$
②

Figure 5.2: Example of finite element based definition of shape determining design variables

design change (out-of-plane)



regularization movement (in-plane) for maintaining mesh quality

Figure 5.3: Design modification and regularization movement

Figure 5.2 shows an example of a finite element mesh for a curved surface, where at three nodes design parameters for shape optimization are visualized. For this purpose, at each node i an attendant coordinate system with the base vectors $\boldsymbol{\theta}_{1,i}, \boldsymbol{\theta}_{2,i}, \boldsymbol{\theta}_{3,i}$ is defined, with $\boldsymbol{\theta}_{1,i}$ and $\boldsymbol{\theta}_{2,i}$ being tangential vectors and $\boldsymbol{\theta}_{3,i}$ being the normal vector to the surface. The coordinates $t_{3,i}$ in normal directions $\boldsymbol{\theta}_{3,i}$ are considered design parameters for shape modification. At a regular node in the interior of the surface, a tangential movement of the node affects the mesh quality, but not the shape. Therefore, tangential coordinates $t_{1,i}$ and $t_{2,i}$ are adapted by secondary arguments to guarantee good mesh quality. Regularization techniques for shape equality control in shape optimization were developed by eg. Firl, Wüchner, Bletzinger [54], Bletzinger et al [29], Stavropoulou, Hojjat, Bletzinger [121], Hojjat, Stavropoulou, Bletzinger [72] or Scherer, Denzer, Steinmann [110] among others. Figure 5.3 shows a one-node out-of-plane design change and the related in-plane regularization movements.

Knowledge about surface normal directions allows to compute the sensitivity p_i of a response function f with respect to the shape modification at the considered node i by computing the total differential of the response in direction $\boldsymbol{\theta}_{3,i}$:

$$p_i = (\nabla f)^T \cdot \boldsymbol{\theta}_{3,i} \tag{5.1}$$

Here, ∇f is to be understood as the derivative of f with respect to the global Cartesian coordinates x of node i. The term p_i is identified as the discrete equivalent at node i of the continuous shape derivative. Mathematical literature about computation of shape gradients and shape calculus can be found in Delfour, Zolésio [47], Haslinger, Mäkinen [69] or Sokolowski, Zolésio [118].

5.3 Shape and design control

5.3.1 Theory of shape control in node based vertex morphing[2]

Goal of this section is to present the underlying theory of the node based vertex morphing method using discrete nodal parameters. On this way, we start from a continuous formulation and will discretize later on.

We consider a surface Ω in space and related surface coordinates $\xi \in \mathbb{R}^2$ which eventually are called material coordinates as well. The surface geometry is given by the spatial coordinates $x = x(\xi)$ in $\mathbb{R}^3$. The field $t_3 \in \mathbb{R}$ is the field of coordinates in normal direction at every surface point ξ. The normal defines the direction of shape evolution at surface point ξ. Additionally, we consider a control field $s(\xi) \in \mathbb{R}$. The normal coordinate t_3 at a surface point ξ_i is linked to s through the kernel filter function $F(\xi, \xi_i)$ and the convolution integral over the surface Ω:

[2] Parts of this section have been pre-published in Masching, Bletzinger [90]

$$\begin{aligned} t_3(\xi_i) &= \int_\Omega F(\xi,\xi_i)\cdot s(\xi)\,d\xi \\ \int_\Omega F(\xi,\xi_i)\,d\xi &= 1 \end{aligned} \tag{5.2}$$

The shape optimization is driven by the manipulation of the control field s. To that end we define the shape optimization problem as:

$$\begin{aligned} & min\ f(s,u) \\ & u = u(s) \ \text{ from } \ S(s,u) = 0 \end{aligned} \tag{5.3}$$

where f is the objective as a function of the control field s and u which are the fields of state variables, e.g. the displacement fields in $\mathbb{R}^3$, and S is the state equation. Eventually, the optimization problem is enhanced by further constraints. Regarding the actual goal of explaining the shape control techniques and for the sake of simplicity they do not have to be considered.

The derivative of the objective f with respect to the design control field s is determined by following chain rule of differentiation:

$$\frac{df}{ds} = \int_\Omega \frac{df}{dt_3}\cdot\frac{dt_3}{ds}d\xi = \int_\Omega p\cdot\frac{dt_3}{ds}d\xi \tag{5.4}$$

where the shape derivative p is the derivative of the objective with respect to the normal coordinates considering shape variations in the surface normal direction. Furthermore, evaluation of $\frac{df}{ds}$ at position ξ_i yields

$$\frac{df}{ds}(\xi_i) = \int_\Omega p(\xi)\cdot F(\xi_i,\xi)\,d\xi \tag{5.5}$$

We observe that derivatives are treated with the transpose filter operation as compared to the filtering of the normal coordinates t_3. Of course, in the case of symmetric filter functions it holds

$$F(\xi_i,\xi) = F(\xi,\xi_i) \tag{5.6}$$

Typically, a piecewise linear hat-function of C_0-continuity is used as a filter kernel function $F(\xi, \xi_i)$. This function can be written in the form

$$F(\xi, \xi_0, R) = \frac{1}{R^2} \cdot \begin{bmatrix} 0 & \xi < \xi_0 - R \\ \xi + (R - \xi_0) & \xi_0 - R \leq \xi < \xi_0 \\ -\xi + (R + \xi_0) & \xi_0 \leq \xi < \xi_0 + R \\ 0 & \xi \geq \xi_0 + R \end{bmatrix} \tag{5.7}$$

where R is a parameter set by the user, defining the filter radius, and ξ_0 is the center position of the filter. Figure 5.4 shows the filter function F for chosen parameters $R = 0.5$ and $\xi_0 = 0.3$.

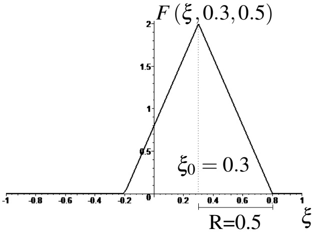


Figure 5.4: Example for filter function F in 1D

In context of finite element analysis, geometry x and design control field s will be discretized. Let us assume that both are discretized using the same mesh and shape functions $N_j(\xi)$. Consequently, within the finite element it holds:

$$s(\xi) = \sum_{j=1}^{n} N_j(\xi) \cdot s_j \tag{5.8}$$

where n is the number of nodes of an element and s_j denotes the discrete control value at node j.

Applying equation 5.8 to equation 5.2 gives

$$t_3(\xi_i) = \sum_{j=1}^{n} \left(\int_{\Omega} F(\xi, \xi_i) \cdot N_j(\xi)\, d\xi \right) s_j \tag{5.9}$$

from which we can derive the discrete filter operator matrix **B** which relates the discrete nodal control numbers s_j and the normal coordinates $t_{3,i}$ at every node:

$$\begin{aligned} t_3(\xi_i) &= t_{3,i} = \sum_{j=1}^{n} B_{ij} \cdot s_j \\ B_{ij} &= \int_{\Omega} F(\xi, \xi_i) \cdot N_j(\xi)\, d\xi \end{aligned} \tag{5.10}$$

For the derivatives it holds:

$$\frac{df}{ds_i} = \frac{df}{dt_{3,j}} \cdot B_{ji} = B_{ij}^T \cdot \frac{df}{dt_{3,j}} = B_{ij}^T \cdot p_j \tag{5.11}$$

Collecting all nodal control parameters and normal coordinates in vectors **s** and $\mathbf{t_3}$, respectively, we get

$$\begin{aligned} \mathbf{t_3} &= \mathbf{B} \cdot \mathbf{s} \\ \frac{df}{d\mathbf{s}} &= \mathbf{B}^T \cdot \mathbf{p} \end{aligned} \tag{5.12}$$

As there are as many discrete parameters for **p** as well as $\mathbf{t_3}$, the matrix **B** is quadratic. For typical filter functions, e.g. the simple hat function, **B** is positive definite and, additionally, for regular element meshes **B** is symmetric. Hence matrix **B** describes a pure parameter transformation, which does not reduce the dimension of the design space. Thus, the original properties of the optimization problem are maintained, such that an optimum determined using the transformation is also an optimum of the original minimization problem.

For the most practical cases with fine meshes and moderately varying mesh density **B** can be approximated to be symmetric.

Considering gradient or quasi Newton methods, it appears that the control parameters **s** may be substituted. The quadratic approximation of f is given by $\tilde{f}$, where $\mathbf{H}_\alpha$ are the matrices of second derivatives with respect to α, which is either the control parameters $\mathbf{s}$ or the normal coordinates $\mathbf{t_3}$:

$$\begin{aligned}\tilde{f} &= f + \frac{df^T}{d\mathbf{s}} \cdot \Delta\mathbf{s} + \frac{1}{2} \cdot \Delta\mathbf{s}^T \cdot \mathbf{H_s} \cdot \Delta\mathbf{s} = \\ &= f + \mathbf{p}^T \cdot \mathbf{B} \cdot \Delta\mathbf{s} + \frac{1}{2} \cdot \Delta\mathbf{s}^T \cdot \mathbf{B} \cdot \mathbf{H_{t_3}} \cdot \mathbf{B}^T \cdot \Delta\mathbf{s}\end{aligned} \tag{5.13}$$

From equation 5.13 together with equation 5.12 the update $\Delta\mathbf{t_3}$ of nodal coordinates can be derived as

$$\begin{aligned}\Delta\mathbf{s} &= -\left(\mathbf{B}^T \cdot \mathbf{H_{t_3}} \cdot \mathbf{B}\right)^{-1} \cdot \mathbf{B}^T \cdot \mathbf{p} \\ \Delta\mathbf{t_3} &= -\mathbf{B} \cdot \left(\mathbf{B}^T \cdot \mathbf{H_{t_3}} \cdot \mathbf{B}\right)^{-1} \cdot \mathbf{B}^T \cdot \mathbf{p}\end{aligned} \tag{5.14}$$

From a practical point of view it appears that the control parameters must not be considered when the method is coded. As equation 5.12 holds, considerations can be reduced to the design update $\Delta\mathbf{t_3}$ and the operator $\mathbf{B}$ for implementation purpose.

Furthermore, on the line of quasi Newton methods, equation 5.14 may be modified as

$$\begin{aligned}&\Delta\mathbf{t_3} = -\alpha \cdot \mathbf{B} \cdot \mathbf{B}^T \cdot \mathbf{p} \\ \text{or }\ &\Delta\mathbf{t_3} = -\alpha \cdot \mathbf{B} \cdot \mathbf{p} = -\alpha \cdot \tilde{\mathbf{p}}\end{aligned} \tag{5.15}$$

where α is a line search factor. Indeed, the normal coordinates $\mathbf{t_3}$ may be updated by means of applying the filter once or twice to the shape derivative $\mathbf{p}$. Since $\mathbf{B}$ is positive definite it is guaranteed that descend directions are generated. Numerical experience shows that both alternatives of equation 5.15 can be used very efficiently for very large problems together with steepest descent or conjugate gradient methods for unconstrained and constrained problems [25] [72]. The second variant of equation 5.15 meets the well-known sensitivity filtering techniques. The method is most efficient together with adjoint methods of sensitivity analysis.

The matrix $\mathbf{B}$ allows for deeper insight into the method. However, it must never be evaluated or stored. Instead, regarding the second variant of equation 5.15 the sensitivity filtering is performed node wise by applying an alternative numerical integration scheme considering derivatives of adjacent nodes within the filter radius R:

$$\tilde{p}_i = \frac{\sum_j F\left(\xi_j, \xi_i\right) \cdot p_j}{\sum_j F\left(\xi_j, \xi_i\right)} \tag{5.16}$$

Indeed, the sensitivity filtering (equation 5.16) is equivalent to techniques which are applied in topology optimization by the SIMP method [20]. Typically, symmetric hat filters are used.

The procedure can be interpreted in different ways. From the mathematical point of view, it is a smoothing operation typically introduced for regularization [103] [77]. In the context of structural shape optimization, this procedure is commonly interpreted and used as a design tool. It gives control over the optimization procedure to the user by choice of the filter radius R as an extra problem parameter. Although the filter cancels out in the course of optimization - which can be seen from equation 5.14 - it improves the numerical properties of the optimization problem by providing smooth geometry updates. On the other hand for non-convex problems, the choice of the filter radius will decide which local minimum will be reached at convergence. This is the intended effect of the filter, which supports the user by exploring the design space, just by simply choosing different filter radii [25] [72].

5.3.2 Examples

5.3.2.1 Convex optimization problem: Generation of catenary curves

The following example applies the presented filtering scheme to a convex problem of shape optimization. The system under consideration is a frame structure, subjected to dead load. Objective of the optimization is minimization of strain energy inside the entire structure, which is a global measure for the flexibility and deformability of the structure (formulations of the response function and the sensitivities can be found in A.1.2). The initial configuration of the system is shown in figure 5.5 with round "corners" joining the vertical and the horizontal members.

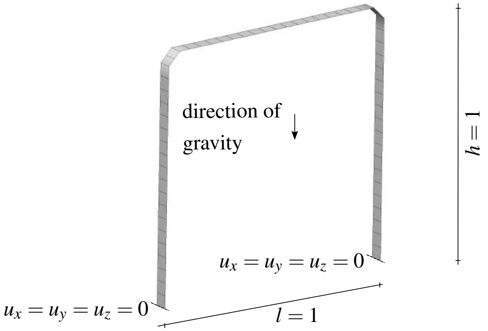


Figure 5.5: System of frame under dead weight

For this optimization problem, the well-known catenary curve represents the optimal shape. In order to ensure a convex optimization problem possessing a unique solution, an equality constraint with respect to the system's mass is added to the optimization problem. Hence the entire optimization problem can be formulated in the form

$$\begin{aligned} &min\ E_{lin} = \frac{1}{2} \cdot \mathbf{u}^T \cdot \mathbf{f_{ext}} \\ &st \\ &m - m_0 = 0 \end{aligned} \tag{5.17}$$

which is solved using the Augmented Lagrangian Method (see section 4.3.5.4).

Optimization is performed using two different filter radii, $R = 0.3$ and $R = 0.6$. The corresponding cross sections of the structure after 10 and 25 optimization steps are compared in figures 5.6 and 5.7. The figures show

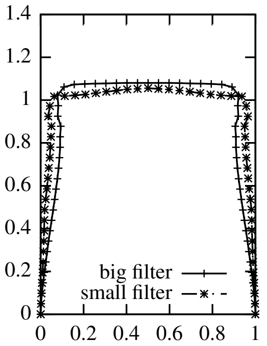


Figure 5.6: Optimization of frame: comparison of designs after 10 steps

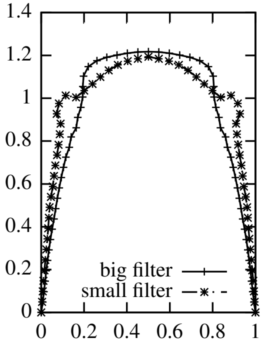


Figure 5.7: Optimization of frame: comparison of designs after 25 steps

that the designs obtained by the different filter radii severely differ from each other, especially at the corners of the frame structure. Figure 5.7 clearly shows that the bigger filter radius accelerates the resolving process of the frame's corners, while the initial local shape of the corners are still present in this design stage for the small filter. However, comparing the converged design for both filter sizes in figure 5.8 reveals that both optimizations result in the same design, which is the expected shape of a catenary curve.

This example confirms that the presented sensitivity filtering technique does not effect the result obtained for convex optimization problems. Nevertheless, the increased smoothness of design updates provided by a larger filter radius led to an increased speed of shape adaptation in the early optimization phase.

5.3.2.2 Non-convex optimization problem: Optimization of a tunnel shell

In the previous example, the filtering technique was applied to a convex problem of shape optimization in order to prove that the chosen filter

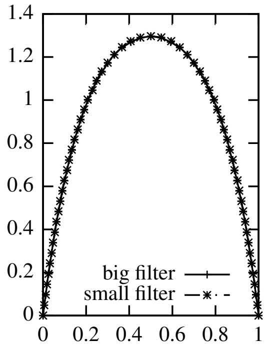


Figure 5.8: Optimization of frame: comparison of designs in converged state (200 optimization steps)

radius does not influence the final design for such problems. Furthermore, equation 5.14 shows that the filter radius cancels out at all for Newton Methods, such that an optimum determined using a filter is an optimum of the initial problem as well, even for non-convex optimization problems. In order to prove this property of the applied filtering scheme, this example considers shape optimization of a snow loaded cylinder shell (figure 5.9) with respect to strain energy. A large Poisson ration of $\nu = 0.45$ is

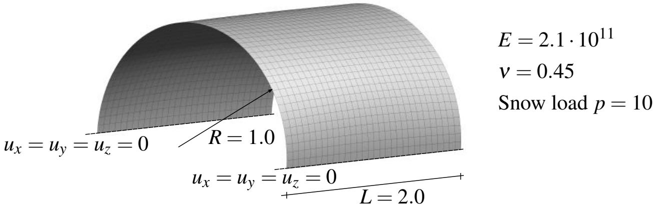


Figure 5.9: System of snow loaded tunnel shell

applied, in order to increase the non-convexity of the optimization problem. Due to Poisson's effect, the free curved edges of the shell undergo a deformation which is larger than deformations observed in the center region. For this reason, gradient based optimization will start modifying the shape of the free edges, while the central and more homogeneous part of the shell is expected to be treated in second rank.

Shape optimization is performed for this problem using filter radii of $R = 0.125$, $R = 0.25$ and $R = 0.5$. The optimized designs are shown in figures 5.10, 5.11 and 5.12. Results reveal the expected behaviour, with shape modifications being dominant at the free edges. This property is distinct especially for the smallest filter radius, where the edge is striving for a catenary shape. An increase of the filter radius emphasizes

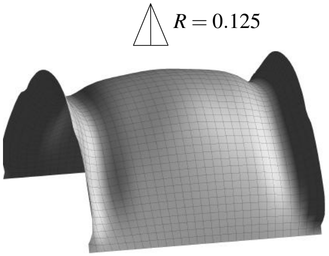


Figure 5.10: Optimized design of tunnel shell, $R = 0.125$

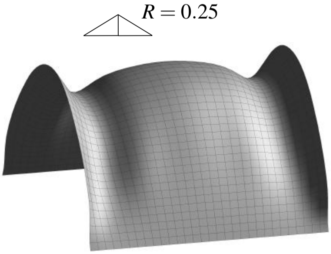


Figure 5.11: Optimized design of tunnel shell, $R = 0.25$

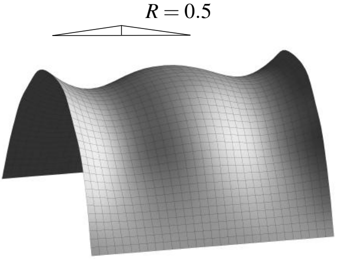


Figure 5.12: Optimized design of tunnel shell, $R = 0.5$

shape changes in the center of the structure, leading to severely different designs, especially for the large filter radius $R = 0.5$.

Despite their differences with respect to shape, all three designs are claimed to be optima. In order to check for optimality, the optimality criteria for the unconstrained optimization problem $\nabla f = \mathbf{0}$ is evaluated. Figure 5.13 shows the development of the norm of the objective gradient over optimization history. It has to be mentioned that in this figure the "pure" gradient information with respect to the design parameters $\mathbf{t_3}$ without any filtering is considered. The presented data shows that all three optimizations, regardless which filter size is used, are converging to $||\nabla f||_2 = 0 \Leftrightarrow \nabla f = \mathbf{0}$, and all three designs can be considered optimal solutions with respect to the initial optimization problem not including any filtering.

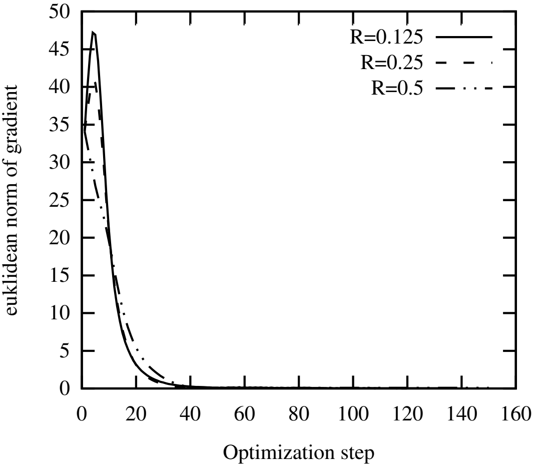


Figure 5.13: Tunnel shell: Development of gradient norms

Although the norms of the gradients do not converge to machine precision, the remaining gradients in orders of magnitude of $||\nabla f||_2 \approx 10^{-2}$ respectively $||\nabla f||_{inf} \approx 10^{-3}$ can be considered to define a local optimum in context of a finite element based sensitivity analysis. The topic of finite element based sensitivity analysis will be discussed in the upcoming section 5.4.

In summary, the presented designs can be considered optimal solutions of the pure optimization problem, whereat the filter only was used in order to guide the optimization algorithm into the radius of convergence of the corresponding optimum.

5.4 Finite element based sensitivity analysis

The field of sensitivity analysis addresses the issue of computing the effect of the variation of an input parameter onto an output parameter. In gradient based optimization, sensitivity analysis is used in order to compute gradients of response functions with respect to the variations of design parameters. Sensitivity analysis has been an intense field of research for mathematicians and engineers over the last decades. While this thesis represents an engineer's point of view onto the topic, more mathematical literature can be found under Choi, Kim [39] [40] or Haug, Choi, Komkov [70], for example.

5.4.1 Different methods of sensitivity analysis

Different types of sensitivity analysis can be distinguished. The most simple form of sensitivity analysis is the *global finite difference method*, which approximates the gradient information of the response by a relative difference of two evaluations of the response function

$$\frac{\partial f}{\partial s_i} \approx \frac{\Delta f^{(fwd)}}{\Delta s_i} = \frac{f(\Delta \mathbf{s} + \mathbf{s}, \mathbf{u}) - f(\mathbf{s}, \mathbf{u})}{\varepsilon} \tag{5.18a}$$

$$\frac{\partial f}{\partial s_i} \approx \frac{\Delta f^{(cent)}}{\Delta s_i} = \frac{f(\Delta \mathbf{s} + \mathbf{s}, \mathbf{u}) - f(-\Delta \mathbf{s} + \mathbf{s}, \mathbf{u})}{2 \cdot \varepsilon} \tag{5.18b}$$

$$\Delta \mathbf{s} = \left\{ \begin{array}{ll} \Delta s_j = 0 & j \neq i \\ \Delta s_j = \varepsilon & j = i \end{array} \right\} \tag{5.18c}$$

where $\Delta\mathbf{s}$ is a vector which applies a finite disturbance ε to the design parameter for which the gradient value is evaluated. The equation above shows two different types of finite differences. In equation 5.18a, a so-called *forward finite difference* is used, which uses the evaluation of f at the point of consideration $\mathbf{s}$, and just one disturbance in forward direction is used. In contrast, in equation 5.18b a *central finite difference scheme* is applied, which uses two disturbed evaluations of f, one in forward and one in backward direction. The different methods will be further discussed in section 5.4.2.

The advantage of the global finite difference is that it does not contain any analytical derivative information, which makes it very easy to implement and quite insusceptible to implementation errors. The drawback is that a huge number of response function evaluations is necessary, which makes it impracticable for serious applications, but global finite differencing can be used as a reference solution.

For more efficient methods of sensitivity analysis, the derivatives of the response functions are determined analytically and implemented as individual functions of $\mathbf{s}$ and $\mathbf{u}$. Depending on how far the analytic approach is followed up, one can further distinguish between *analytic sensitivity analysis* and *semi-analytic sensitivity analysis*. In analytic sensitivity analysis, the entire sensitivity analysis is performed analytically, whereat in the semi-analytic approach the analytic procedure is performed until to a certain extent, and remaining parts of the sensitivity equation are approximated by a finite difference scheme. In this thesis, semi-analytic sensitivity analysis is used, which will be further explained and discussed in section 5.4.2.

Another classification of sensitivity analysis methods can be done by grouping them in *variational sensitivity analysis* and *discrete sensitivity analysis*. The difference between these groups is the order of the proceeding steps. While in discrete sensitivity analysis, the response function is discretized in the first step and then the derivatives with respect to discrete parameters are evaluated, the variational method proceeds the other way round. In this case, the derivatives are computed in a continuous form and afterwards a discretization method such as finite elements is applied to the continuous derivates. Within this thesis, discrete formulations are considered only. Further information about variational sensitivity analysis can be found in Barthold, Stein [13], Ibrahim, Tiwari [76] or Daoud [46].

5.4.2 Semi-analytic sensitivity analysis

5.4.2.1 Computing semi-analytic sensitivities

As already mentioned in the previous section, a semi-analytic approach is used in order to compute sensitivities within this thesis. In this approach, response functions are differentiated analytically until element level. But instead of now deriving element entities, such as stiffness or internal forces, analytically, a finite difference scheme is used in order to approximate these derivatives.

This approach will be explained considering the sensitivity of the determinant of a system stiffness matrix **K** with respect to the design parameters **s**. According to Jacobi's formula [18], the sensitivity with respect to design variable s_i can be written as

$$\frac{\partial det\left(\mathbf{K}\left(\mathbf{s}\right)\right)}{\partial s_i} = det\left(\mathbf{K}\left(\mathbf{s}\right)\right) \cdot trace\left(\mathbf{K}\left(\mathbf{s}\right)^{-1} \cdot \frac{\partial \mathbf{K}\left(\mathbf{s}\right)}{\partial s_i}\right) \tag{5.19}$$

assuming our system to be sufficiently supported such that $\mathbf{K}^{-1}$ exists. The system stiffness matrix **K** is a global variable, which is generated by assembling over all element matrices $\mathbf{k_e}$ in the computational domain

$$\mathbf{K} = \underset{\text{all}}{A}\ \mathbf{k_e} \tag{5.20}$$

with A being the assembly operator. At this point, the derived stiffness matrix $\frac{\partial \mathbf{K}}{\partial s}$ behaves differently. As the modification of a design parameter, such as a nodal position, influences only a small number of neighboring finite elements, just the stiffness of these elements will be changed, while

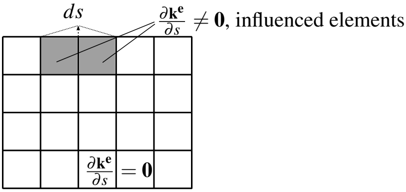

ds
$\frac{\partial \mathbf{k^e}}{\partial s} \neq \mathbf{0}$, influenced elements
$\frac{\partial \mathbf{k^e}}{\partial s} = \mathbf{0}$

Figure 5.14: Elements influenced by a design modification ds, using the example of an in-plane design modification

the main part of the finite element domain remains unaffected. Figure 5.14 illustrates the located effect of such a design modification on a plate in plane stress/strain, so ds is an in-plane design modifcation. Accordingly, it is sufficient to assemble over a quite small number of influenced finite elements when computing the derived stiffness matrix, as for the unaffected elements the derived element stiffness $\frac{\partial \mathbf{k_e}}{\partial s}$ turns out to be zero. Hence the assembling procedure for the derived stiffness matrix can be reduced to the small quantity of influenced elements *infl*:

$$\frac{\partial \mathbf{K}}{\partial s} = \underset{\text{infl}}{A} \frac{\partial \mathbf{k_e}}{\partial s} \tag{5.21}$$

So far, all computations have been performed analytically, and equations 5.19, 5.20 and 5.21 do not contain any approximations. The idea of the semi-analytic sensitivity is, that at this point the evaluation of analytical derivatives of element specific quantities like $\frac{\partial \mathbf{k_e}}{\partial s}$ is avoided by replacing them using a finite differencing scheme according to equations 5.18a or 5.18b

$$\frac{\partial \mathbf{K}}{\partial s} \approx \underset{\text{infl}}{A} \frac{\Delta \mathbf{k_e}}{\Delta s}^{(fwd)} \approx \underset{\text{infl}}{A} \frac{\Delta \mathbf{k_e}}{\Delta s}^{(cent)} \tag{5.22}$$

In contrast to the global finite difference approach, it is not necessary to re-evaluate the entire objective function in the disturbed state $\Delta\mathbf{s} + \mathbf{s}$, but only the stiffness matrices of the elements influenced by the design variable.

Using equations 5.19, 5.21 and 5.22, the final formulations of the analytic (equation 5.23) and the semi-analytic sensitivities (equation 5.24) can be formulated, whereby (x) denotes the chosen finite differencing scheme.

$$\frac{\partial det\left(\mathbf{K}\left(\mathbf{s}\right)\right)}{\partial s} = det\left(\mathbf{K}\left(\mathbf{s}\right)\right) \cdot trace\left(\mathbf{K}\left(\mathbf{s}\right)^{-1} \cdot \underset{\text{infl}}{A} \frac{\partial \mathbf{k_e}}{\partial s}\right) \tag{5.23}$$

$$\frac{\partial det\left(\mathbf{K}\left(\mathbf{s}\right)\right)}{\partial s} \approx det\left(\mathbf{K}\left(\mathbf{s}\right)\right) \cdot trace\left(\mathbf{K}\left(\mathbf{s}\right)^{-1} \cdot \underset{\text{infl}}{A} \frac{\Delta \mathbf{k_e}}{\Delta s}^{(x)}\right) \tag{5.24}$$

The usage of the semi-analytic method has several advantages towards the pure analytical approach:

- The implementation of analytic derivatives of element based data can be omitted. These derivatives may become very tedious, especially when complex element formulations like shell elements with hybrid stress formulation are used.
- Finite differencing schemes according to equation 5.22 can be used for any element formulation, whereby any element which is available for simulation directly can be used for optimization purpose without additional coding.
- Any entity of the finite element formulation, such as shape, thickness or material orientation, immediately can be used as a design parameter without specific implementations on element level.
- For most element formulations, the semi-analytic approach turns out to be faster than the complete analytical sensitivity formulation. This is because evaluating the element stiffness twice and applying a finite differencing scheme is less costly, from the numerical point of view, than following the product and chain rules of differentiation in order to determine the analytic sensitivity.

5.4.2.2 Numerical efficiency

In order to judge about the efficiency of the semi-analytic approach, we consider a very simple four-noded plain stress element with linear kinematics. The stiffness matrix of this element can be writen in the well-known form

$$\mathbf{k_e} = \int\limits_V \mathbf{B}^T \cdot \mathbf{D} \cdot \mathbf{B} \cdot det\left(\mathbf{J}\right)\ dV \tag{5.25}$$

Taking the analytical derivative of the stiffness matrix with respect to a design variable s, equation 5.25 extends to a three summand expression due to the product rule of differentiation:

$$\begin{gathered} \frac{\partial \mathbf{k_e}}{\partial s} = \int\limits_V \frac{\partial \mathbf{B}}{\partial s}^T \cdot \mathbf{D} \cdot \mathbf{B} \cdot det\left(\mathbf{J}\right) + \\ \mathbf{B}^T \cdot \mathbf{D} \cdot \frac{\partial \mathbf{B}}{\partial s} \cdot det\left(\mathbf{J}\right) + \mathbf{B}^T \cdot \mathbf{D} \cdot \mathbf{B} \cdot \frac{\partial det\left(\mathbf{J}\right)}{\partial s}\ dV \end{gathered} \tag{5.26}$$

Due to this extension, the computation of the derived stiffness is approximately three times as costly as computing the stiffness itself. As the forward and the central finite differencing scheme only require two evaluations of the stiffness matrix, we expect the semi-analytic sensitivity computation to save about 33% of the computational time compared to the analytic approach. In order to get a unaltered comparison of the real cpu-times, a separate C++ implementation of the stiffness matrix, its derivative and the finite differencing schemes is used. The needed cpu-times in order to compute $100,000$ derivatives of element stiffness matrices are presented in figure 5.15 [3]. As we can see, the semi-analytic approach is faster by 25% compared to the analytic approach. This saving of computational time is smaller than the initial expectation of 33%, but this expectation neglected the time consumed by evaluating the finite difference as well as the memory allocation overhead caused by the higher number of function calls in the semi-analytic procedure.

[3] Serial computation on an Intel i3 processor

Method	cpu-time $[ms]$	rel. time
Analytic	$1,688$	100%
Semi-analytic	$1,275$	75%

Figure 5.15: Comparison of cpu-times for computing $100,000$ derivatives of stiffness matrices

The overall cpu-times shown in figure 5.15 are relatively small, which is due to the simplicity of the considered element. Nevertheless, the observed saving of 25% is significant and will even increase with an increasing element complexity, as well as the overall cpu-time will. From this point of view, the semi-analytic approach is an essential contribution to efficient sensitivity analysis of large structures using high-end elements, as e.g. those with additional internal variables to treat the various locking effects.

5.4.2.3 Accuracy and optimal disturbance

The main point of criticism concerning semi-analytic sensitivity analysis is its non-exactness [98]. As element data derivatives are approximated using a finite difference, the approximation implies either an approximation or a round-off error, depending on the size of the perturbation parameter ε.

In this context, different sources of error have to be distinguished. The most severe error is caused by a numerical slipping off of the reference element coordinate system due to the finite difference. Due to the perturbation of the design, the finite difference approximation subtracts two stiffness matrices which have been computed with respect to different reference configurations of the element, resulting in an approximated derived stiffness matrix $\frac{\Delta \mathbf{K}}{\Delta s}$ that violates the rigid rotation conditions $\mathbf{\Phi}^T \cdot \frac{\Delta \mathbf{K}}{\Delta s} \cdot \mathbf{\Phi} = 0$, where $\mathbf{\Phi}$ is an eigen vector of the stiffness matrix $\mathbf{K}$, describing a rigid body rotation of the considered element [12]. Bletzinger et al [26] showed that this effect mainly occurs for elements with complex kinematics like Kirchhoff formulations, and presented a method of exact semi-analytic sensitivities correcting this deficit. For this purpose, a modified approximated derived stiffness matrix $\frac{\Delta \mathbf{K}^*}{\Delta s}$ is computed by adding correction terms curing the violation of rigid body conditions. Equation 5.27 shows the correction, where $\mathbf{\Phi}_r$ is a rigid body rotation vector and n_r is the total number of possible rigid body rotations.

$$\left[\mathbf{\Phi}_r^k\right]^T \cdot \underbrace{\left(\frac{\Delta \mathbf{K}}{\Delta s} + \sum_{i=1}^{n_r}\sum_{j=1}^{n_r} a_{i,j} \cdot \mathbf{\Phi}_r^i \cdot \left[\mathbf{\Phi}_r^j\right]^T\right)}_{\frac{\Delta \mathbf{K}^*}{\Delta s}} \cdot \mathbf{\Phi}_r^l = 0$$

$$\forall \;\; k,l \in \{1,...,n_r\} \tag{5.27}$$

with

$$a_{i,j} = -\frac{\left[\mathbf{\Phi}_r^i\right]^T \cdot \frac{\Delta \mathbf{K}}{\Delta s} \cdot \mathbf{\Phi}_r^j}{\left(\left[\mathbf{\Phi}_r^i\right]^T \cdot \mathbf{\Phi}_r^i\right) \cdot \left(\left[\mathbf{\Phi}_r^j\right]^T \cdot \mathbf{\Phi}_r^j\right)}$$

Applying this corrections, the remaining error in the semi-analytic sensitivity reduces to the approximation error contributed by the finite differencing scheme. This error is directly related to the finite disturbance ε.

The following convergence study investigates the error in the approximated sensitivity for a plain stress element formulation, comparing the forward and the central finite difference scheme with and without the rigid body correction according to equation 5.27. The results are displayed in figure 5.16, where the error is measured by the Frobenius norm of the difference between the analytical derivative of $\mathbf{k_e}$ and the finite difference approximation. Figure 5.16 presents that for $\varepsilon > 10^{-5}$ both finite difference schemes show a constant decrease of error with the expected orders of accuracy. The forward finite difference shows an order of accuracy of 1 while the central finite difference method possesses an order of accu-

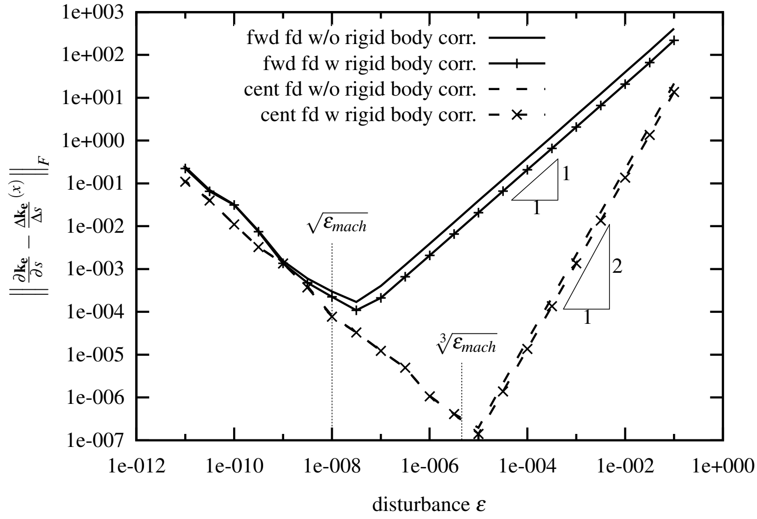


Figure 5.16: Convergence of error in semi analytic sensitivity analysis

racy of 2 [4]. For both methods, a critical value of ε exists. If this value is undercut, a decrease of ε leads to an increasing error. In this range, round-off errors dominate the result, as the differences computed in the finite difference scheme go to zero, which reduces the accuracy of the resulting floating point number [45]. Depending on the chosen finite differencing scheme and the machine precision ε_{mach}, optimal values for ε can be determined providing the best trade-off between approximation and round-off error. Assuming 64-bit floating point numbers with a machine precision of $\varepsilon_{mach} = 10^{-16}$, optimal disturbance values arise according to figure 5.17.

Finite difference scheme	ε_{opt}
Forward	$\varepsilon_{opt}^{(fwd)} = \sqrt{\varepsilon_{mach}} = 10^{-8}$
Central	$\varepsilon_{opt}^{(cent)} = \sqrt[3]{\varepsilon_{mach}} \approx 4.6 \cdot 10^{-6}$

Figure 5.17: Optimal disturbance values for finite differencing schemes

Figure 5.16 also demonstrates that the rigid body correction leads to slightly improved results for both finite difference schemes. For $\varepsilon > \varepsilon_{opt}$, the influence of the rigid body correction is constant and comparatively small to the approximation error of the finite difference method for this type of element.

5.5 Direct and adjoint sensitivity formulation

This chapter focuses on formulating the sensitivities of response functions typically arising from applications in structural optimization. These response functions I usually depend on the design $\mathbf{s}$ and on the state variables $\mathbf{u}$.

$$I(\mathbf{u},\mathbf{s}) \tag{5.28}$$

Taking the derivative of I with respect to a design variable $s_i \in \mathbf{s}$ under consideration of the chain rule of differentiation leads to

$$\frac{dI}{ds_i} = \frac{\partial I}{\partial s_i} + \left[\frac{\partial I}{\partial \mathbf{u}}\right]^T \cdot \frac{\partial \mathbf{u}}{\partial s_i} \tag{5.29}$$

where the derivative of the state variables is not known so far.
The state variables $\mathbf{u}$ are implicitly defined via the state equation $\mathbf{S}$, demanding the structure to be in equilibrium:

$$\mathbf{S}(\mathbf{u}) = \mathbf{0} \tag{5.30}$$

As the state equation has to be fulfilled for any design $\mathbf{s}$, the derivative of $\mathbf{S}$ with respect to all $s_i \in \mathbf{s}$ has to be zero. This allows to solve for the unknown derivatives of the state variables:

$$\frac{d\mathbf{S}}{ds_i} = \frac{\partial \mathbf{S}}{\partial s_i} + \frac{\partial \mathbf{S}}{\partial \mathbf{u}} \cdot \frac{\partial \mathbf{u}}{\partial s_i} = 0 \Rightarrow \frac{\partial \mathbf{u}}{\partial s_i} = \left[\frac{\partial \mathbf{S}}{\partial \mathbf{u}}\right]^{-1} \cdot \left[-\frac{\partial \mathbf{S}}{\partial s_i}\right] \tag{5.31}$$

Inserting equation 5.31 into equation 5.29 leads to

$$\frac{dI}{ds_i} = \frac{\partial I}{\partial s_i} + \left[\frac{\partial I}{\partial \mathbf{u}}\right]^T \cdot \underbrace{\left[\frac{\partial \mathbf{S}}{\partial \mathbf{u}}\right]^{-1} \cdot \left[-\frac{\partial \mathbf{S}}{\partial s_i}\right]}_{\frac{\partial \mathbf{u}}{\partial s_i}} \tag{5.32}$$

which is called the *direct sensitivity* formulation of response function I [39] [40]. It has to be mentioned that the inverse of $\frac{\partial \mathbf{S}}{\partial \mathbf{u}}$ usually is not determined explicitly in large-scale applications. Instead, a factorization scheme like Gauss or Cholesky is applied in order to transform the matrix to a triangular shape, and the inverse matrix-vector multiplication can be performed doing a so-called forward-backward substitution [94].

In the direct formulation, forward-backward substitution representing the derivatives of the state variable with respect to the design variable s_i has to be solved for each design variable, as the right-hand sides of the linear systems $-\frac{\partial \mathbf{S}}{\partial s_i}$ depend on s_i. The intension of *adjoint sensitivity* formulations is to re-formulate equation 5.32 such that the solution of a linear system of equations for each design variables becomes needless. For this purpose, the rear summand of the equation is transformed switching the order of execution in the vector-matrix-vector-product exploiting the following relation:

$$\mathbf{v^T} \cdot \mathbf{M} \cdot \mathbf{w} = \mathbf{w^T} \cdot \mathbf{M^T} \cdot \mathbf{v} \tag{5.33}$$

The transformed version of equation 5.32 reads

$$\frac{dI}{ds_i} = \frac{\partial I}{\partial s_i} + \left[-\frac{\partial \mathbf{S}}{\partial s_i}\right]^T \cdot \left[\frac{\partial \mathbf{S}}{\partial \mathbf{u}}\right]^{-T} \cdot \frac{\partial I}{\partial \mathbf{u}} \tag{5.34}$$

where the Jacobi matrix of the state equation $\frac{\partial \mathbf{S}}{\partial \mathbf{u}}$ is identical with the stiffness matrix **K** in structural mechanics. As the stiffness matrix is symmetric for conservative problems [19], the inverse of it is symmetric as well [123]. Therefore the same operator can be used in the adjoint formulation as in the direct formulation,

$$\frac{dI}{ds_i} = \frac{\partial I}{\partial s_i} + \left[-\frac{\partial \mathbf{S}}{\partial s_i}\right]^T \cdot \underbrace{\left[\frac{\partial \mathbf{S}}{\partial \mathbf{u}}\right]^{-1} \cdot \frac{\partial I}{\partial \mathbf{u}}}_{\Lambda} \tag{5.35}$$

where the variable $\Lambda = \left[\frac{\partial \mathbf{S}}{\partial \mathbf{u}}\right]^{-1} \cdot \frac{\partial I}{\partial \mathbf{u}}$ is called the *adjoint variable*. As the operator matrix for computing the adjoint variable is the same as for

determing the derivatives of the state variables $\frac{\partial \mathbf{u}}{\partial s_i}$ in structural mechanics, these problems are called *self-adjoint*.
In contrast to the direct approach, where the derivative of state variables has to be computed for each single design variable involving the solution of a linear system of equations, the adjoint approach allows to determine the derivatives of I with respect to all design variables by only solving one linear system in order to compute the adjoint variables. From this point of few, the adjoint approach is preferable when optimization problems defined by a large number of design variables, but only involving a small number of response functions, are solved. As soon as the number of response functions approaches the number of design variables, the advantage of the adjoint approach vanishes, as the number of linear solutions being necessary to determine the derived state variables is independent of the number of responses, but the effort to compute the adjoint variables increases linearly with the number of response functions.
In parameter free optimization, as it is performed in this thesis, the number of design variables usually is larger than the number of response functions by several orders of magnitude for large examples. For this reason, an adjoint formulation of the response functions is a key issue to numerical efficiency in the performed optimizations.
More information about adjoint sensitivity formulations can be found in Giles, Pierce [63] or Cao, Li, Petzold [36].

Adjoint sensitivity formulations of frequently used response functions can be found in appendix A.

5.6 Line search procedures

After the sensitivity information has been computed and filtered for all response functions, the final search direction for the actual optimization step is determined. Thereby the search direction might be a conjugated direction in case of a CG-algorithm or gradient information of a Lagrange function in case of a constrained optimization problem, for example. In the next step, a line search is performed along the search direction, as it was briefly introduced in section 4.2.2.2. In contrast to an exact line search, where the optimal step size α is defined by the gradient of f to vanish $[\nabla f(\mathbf{x}+\alpha\cdot\mathbf{s})]=\mathbf{0}$ (see equation 4.5), usually inexact line search methods are applied in numerical optimization, as the objective function is not known analytically. The most simple type of inexact line search is a fixed step method, which prescribes a fixed value for α. The disadvantage of this method is that the chosen step length is not optimal, which increases the number of necessary optimization iterations or might lead to an "overleaping" of the local minimum. Consequently more sophisticated inexact line search procedures should fulfill the requirement that "the step size should not be too large or too small, and there should be a sufficient decrease in the cost function" [8]. Line search methods which try to meet these criteria using one evaluation of the response function only are Armijo's rule [7] or the Wolfe Condition [134]. These methods use a linear extrapolation of the response function in search direction based on the actual design, and demand a minimum decrease of the response function within each optimization step. Another method of inexact line search is the polynomial interpolation method [111]. Thereby the response function is approximated within an interval of acceptable step sizes using a polynomial function, which now can be used as an auxiliary mean for an exact line search. This method requires a larger number of response function evaluations compared to Armijo's rule or the Wolfe condition, as sampling points of the polynomial function need to be determined. Nevertheless, as evaluation points are distributed over the interval of acceptable step sizes, the gained information usually is more precise and reliable than the information obtained by methods using only one evaluation point. So the line search reflects the real behaviour of the response function more precisely.

5.7 Work flow and data management in a object oriented code environment

This section is intended to illustrate the work flow and the data management of a parameter free optimization, as it is implemented in Carat++ (see section 1.4). Figure 5.18 visualizes the connections and the data streams.

The first thing which is conspicuous, is a modular structure of the implementation. The overall optimization framework includes a structural analysis module, which contains the usual finite element tools like classes for nodes, elements, finite element domains or analysis classes managing different types of structural analysis. Figure 5.18 shows these standard finite element components on the right hand side, separated from the optimization module by a dot and dash line. The optimization module can be regarded to be a add-on onto the analysis module, which is connected by pointer access in an object oriented manner. The points of connection are quite limited. The finite element analysis object is controlled by a response function object via a pointer connection, as well as the design variables have pointer access to the underlying finite element object. This could be either a node or an element.

Despite this limited number of intersection points, the optimization module has got the necessary control over the finite element analysis model. Optimization variables are able to modify the design of the finite element model as well as they can query element data, which is essential for the semi-analytic sensitivity analysis, via the pointer access to a finite element or a node. State variables and adjoint variables for a given pseudo load vector are provided by the analysis object directly to the response function object, and so all data necessary for sensitivity analysis can be gathered inside the response function routines. Once sensitivity information is computed, there is only a very small amount of further communication necessary between analysis and optimization module in the following

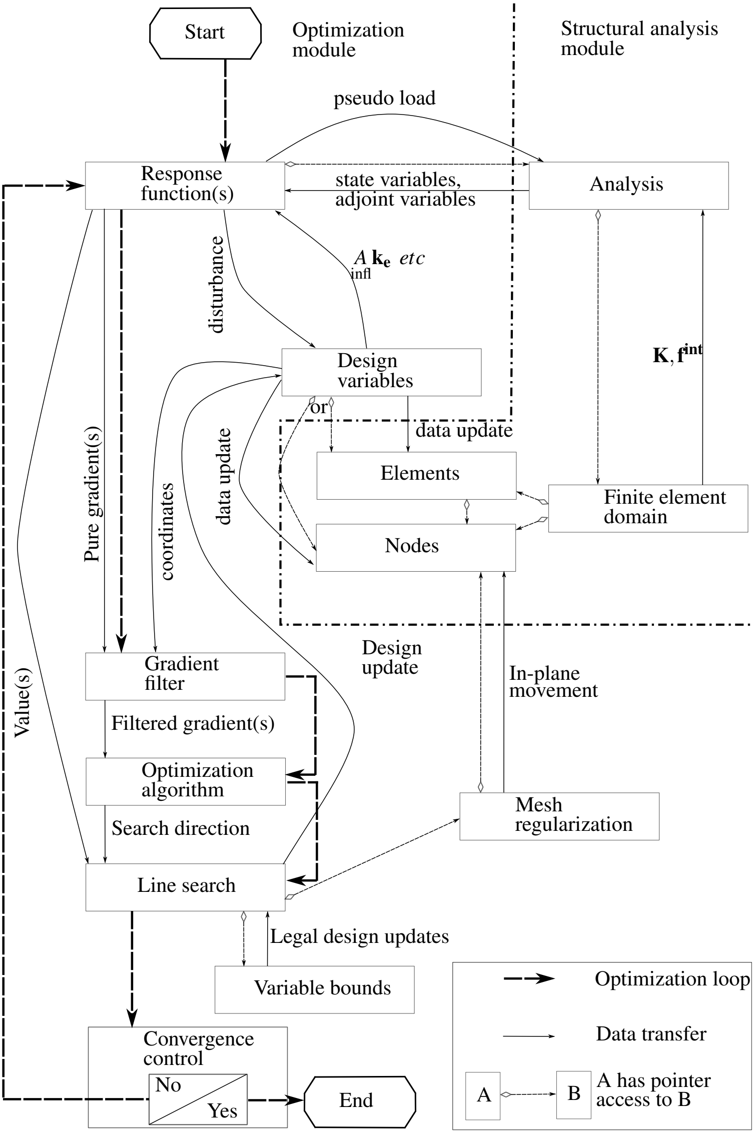


Figure 5.18: Data mangement chart

steps. Pure sensitivity information of the response function, or of several response functions in case of a constrained or multi-objective problem, is passed to the filtering object, which performs the filtering operation described in section 5.3 on the discrete pure sensitivity data. The resulting filtered gradient data is pushed forward to the mathematical core routine of the optimization algorithm, which determines the search direction for the actual optimization step, based on the actual gradient information of all response functions plus some variables coming from optimization history, like Lagrange multipliers or previous search directions. Once the search direction is determined, the line search can be applied. Thereby inexact line search methods as presented in section 5.6 are used. Having computed a step length factor α, the final design update is applied to the design variables, which, for their part, update the finite element model. The quantity of legal design updates may be restricted by variable bounds, which prevent design variables from assuming arbitrary values. By this mean, unphysical or inpracticable designs can be avoided. The line search methods take care of these bounds and correct illegal design updates correspondingly.
The thicker dashed line in figure 5.18 shows the optimization loop. Here the steps

- Gradient computation
- Gradient filtering
- Determination of search direction
- Line search

are repeated until convergence of the optimization is achieved. Therefore, usually convergence criteria based on

- a minimum decrease of the objective function or
- a minimum step length obtained in the line search

obtained in the last optimization iteration are applied, instead of focusing on a gradient based criterion like equation 4.3 or 4.20.

5.8 Summary

In this chapter, the parameter free optimization approach has been presented as a efficient parametrization method for treatment of structural optimization problems. In combination with semi-analytic and adjoint sensitivity computation, it is excellently applicable for fine discretized models of industrial relevance. In the upcoming chapter, these methods and algorithms will be applied in order to generate optimal shapes for efficiently deformable shell structures, being a key issue to actuated systems.

CHAPTER 6

Hingelike mechanisms of shell structures

6.1 Classification of kinematic mechanisms

In order to discuss structures which perform a maximum of movement for a limited actuator input, it is important to investigate kinematic mechanisms of structures, allowing these structures to undergo large deformations while representing a minimum of mechanical resistance. Figure 6.1 presents typical kinematic mechanisms for different types of structures.

For beam structures, nodal hinges allowing the connected parts to undergo independent rotations can be used, which enables the structure to move by producing a minimum of strain energy or no strain energy at all, when a kinematic system is considered. Considering plates, an entire plate can be divided into sub-plates using line hinge mechanisms, which allow each sub-plate to perform a rigid body movement. These line hinges are directly related to ultimate load analysis of plates using yield line theory [78]. Yield line theory assumes a constant yield moment acting along all yield lines in the ultimate load state, and that elastic deformations can be neglected. The ultimate load state is characterized by a patch of sub-plates, each deforming in a rigid body state, whereat the entire internal

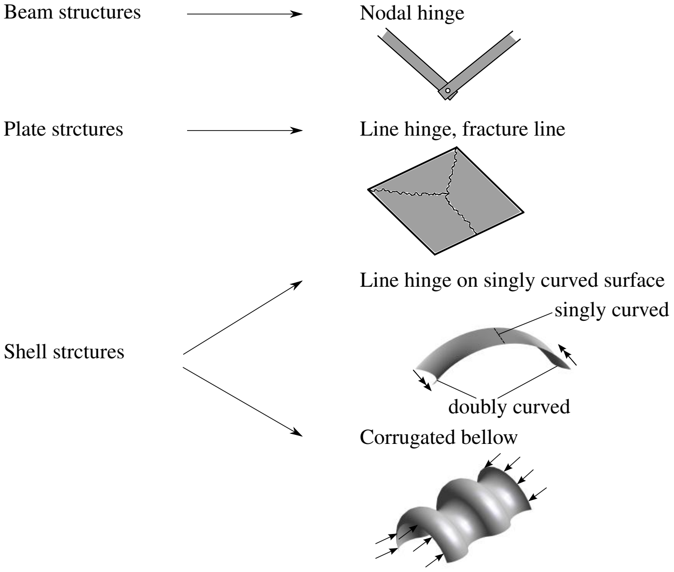


Figure 6.1: Hingelike mechanisms for different types of structures

work inside the system is generated by the yield moments along the yield lines. Due to this rigid body assumption, adjacent sub-plates intersect in straight lines. Therefore the resulting yield line along the intersection is also straight, as long as distributed loads and angled plates are considered [66]. Considering shells, the situation becomes more complicated. The idea of a straight yield line or line hinge offering maximum movability cannot be applied to arbitrarily curved shells. Straight line hinges allowing an almost zero-stress movement of connected shell parts only can be found for singly curved shell parts, where the straight generator line acts as a line hinge. The existence of such straight hinge lines and the corresponding mono-axial curvature leads to highly flexible shell structures. This high flexibility can be observed for example in the phenomenon of ovalizing cylinders, where a singly curved shell performs almost inextensional deformations, and load is carried by a pure state of bending [107] [35]. Figure 6.2 shows the ovalizing of a pinched cylinder.

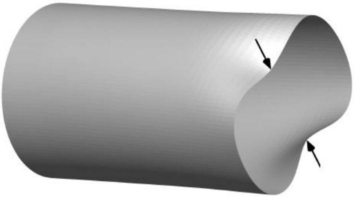


Figure 6.2: Ovalizing of a pinched cylindrical shell

Another shell design offering high flexibility is the corrugated bellow design. It is quite frequently used in industrial applications for shell structures which are mainly effected by normal forces and require high flexibility. The working principle is based on a change of load carrying behaviour, switching from a membrane stress state to a bending dominated state inside the corrugations. Figures 6.3 and 6.4 show some typical applications of corrugated bellows. Although they are mainly used as a secondary structure preventing the main structure from suffering dam-

Figure 6.3: Corrugated bellow in an articulated bus (source of picture: [89])

Figure 6.4: Corrugated bellow as compensator in a hydraulic line (source of picture: [119])

ages, the principle also can be used for primal structures carrying significant load.

This chapter discusses the challenging task of finding ideal shapes of shell structures providing optimal kinematic properties using the parameter free shape optimization approach. Hereby, a shell formulation based on linear shear deformable Reissner-Mindlin theory is used. In this context, different types of shell structures are considered, leading to line hinges or corrugated bellow designs.

6.2 Generation of kinematic mechanisms using optimization

6.2.1 Hinge line pattern of a square plate

In section 6.1, line hinges were introduced as an optimal kinematic mechanism for plates and for certain types of shells. This example compares hinge line patterns motivated by yield line theory [78] to hinge lines generated by optimization. For this purpose, a square plate which is hinge supported at three edges and loaded by a uniform snow load is considered. Optimization is performed with respect to a maximization of strain energy, whereat the plate thickness is variable. Figure 6.5 shows the system and the resulting thickness distribution. In order to obtain a localized hinge line pattern, an equality mass constraint was added to the optimization problem as well as upper and lower bounds for allowable plate thicknesses. Figure 6.5 shows Y-shaped hinge lines, dividing the plate into three sub-slabs. Computations based on yield line theory deliver an identical shape of yield lines, where the intersection of the three yield lines is located $0.65 \cdot L$ from the top edge of the plate [132]. This value is represented by the optimization result very well.

Figure 6.5 also shows the deformation pattern of the optimized design. Here the rigid body deformations of the sub-slabs can be observed quite

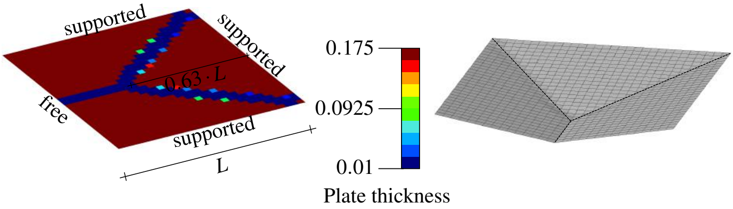


Figure 6.5: Line hinge pattern obtained by optimization (left) and deformation of optimized design (right)

nicely, leading to a maximized strain energy and correspondingly to a maximized deflection of the entire structure.

This example shows that hinge lines providing maximum deflection of plates are identical with results derived from yield line theory. In proceeding steps, these considerations will be transferred to curved structures, and the focus will be put on modifying the shapes of shell structures in order to obtain optimal kinematic properties.

6.2.2 Cylinder roof segment

This example considers a singly curved roof structure, which spans in direction of the generator line. The structure is fixed supported at both curved edges, while the straight edges are unsupported. A line load is acting along the center line of the roof (see figure 6.6). The roof is shape optimized, whereat the objective is to maximize the vertical deflection u_{cent} at the center of the roof.

Due to the loading and the support conditions, the investigated structure is exposed to high bending moments. In contrast to the ovalizing cylinder problem shown in figure 6.2, here the bending moment is acting perpendicular to the shell's generator line. Thus the bending stiffness opposed by the shell is quite high due to its curvature. Thus, the a priori assumed optimal design is expected to show areas where the curvature of the shell

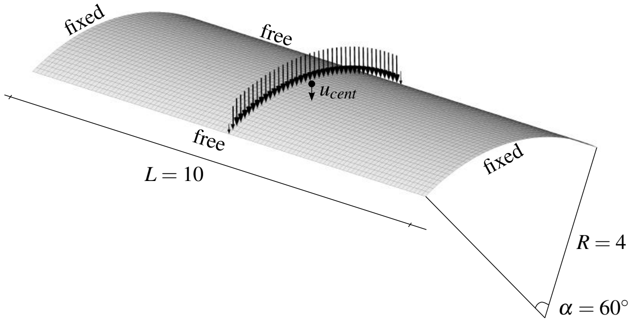


Figure 6.6: System of cylinder roof segment

is reduced or completely removed. These areas will act as line hinges, localizing the bending effects and allowing the main part of the shell to undergo an almost stress free deformation, as it could be investigated for the plate structure in section 6.2.1. In order to allow a stress free deformation, the structure has to be equipped with three line hinges in order to act similar to a kinematic three-hinged frame.

Optimization progress is presented in figure 6.7. The first intermediate result (Stage A) shows main changes at the supports. Here the curvature is severely reduced, generating horizontal support lines. As the support conditions are formulated as fixed but not clamped supports, a straight arrangement of the supported nodes acts as a real hinge, where the shell can perform a free rotation. For this reason, design modifications are more significant at the supports than in the central area of the roof in the early phase of optimization. In the central area, the stiffness of the structure is not only determined by its shape, but also by the bending stiffness due to the shell thickness, so here the shape gradients of the response function are small compared to the support region, resulting in reduced design updates. When optimization proceeds (Stage B), the distribution of shape gradients over the structure changes, and a line hinge develops in the area

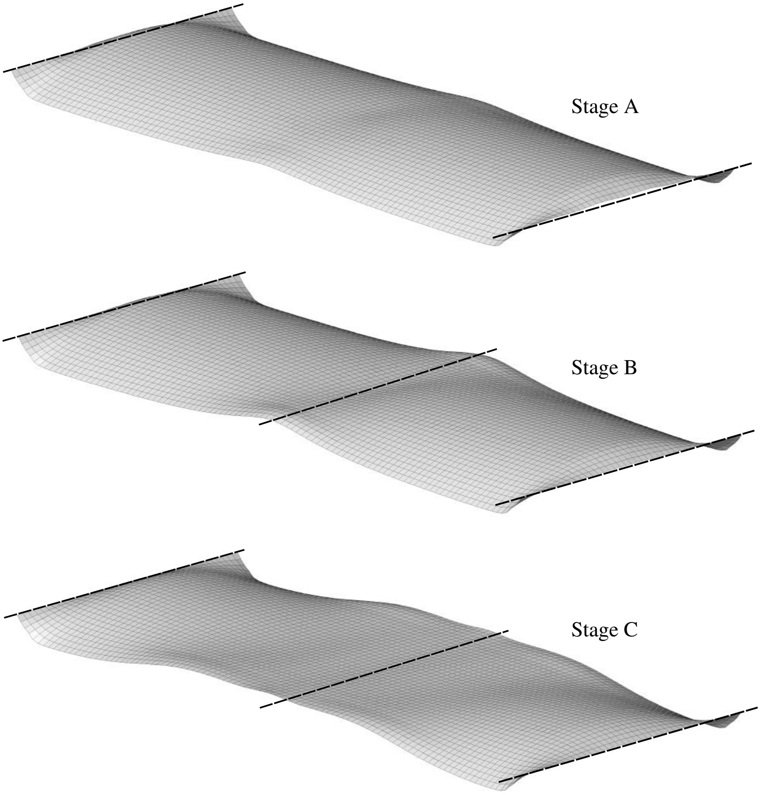


Figure 6.7: Optimization progress of cylinder roof segment

of load application, while the shape modifications at the supports slow down. In the converged configuration (Stage C), the central line hinge starts to spread in span direction to a plane plateau, while the line hinges at the supports stay localized. This again can be explained by the bending effect inside the structure. As there is no bending moment at the supports, a spread of the flat region would not lead to a significant softening of the structure. In contrast, the flat plateau in the central area reduced the stiffness of the structure in a region where it is affected by bending, which increases flexibility.

In a different optimization setup, the structure is optimized applying variable linking. For this purpose, design variables are linked in hoop direction such that the related nodes undergo identical design modifications. As a consequence, the initial curvature of the cylinder is conserved in hoop direction, and an increased flexibility of the structure only can be achieved by the insertion of corrugations. Optimization is performed using a mass constraint, limiting the increase of the system mass to 5%. In figure 6.8, three optimized roofs obtained by using different filter radii are compared. The designs show that for this problem the chosen filter radius has a minor influence onto the final shape of the structure. For all three cases, corrugations have been generated at the supports and in the center of the roof each, whereat the shape and especially the amplitude of the corrugations are governed by the mass constraint. For this bending dominated problem, optimization of shape and location of the corrugations seems to be a convex problem, where different filter radii lead to the same optimal design, as it was discussed in section 5.3.

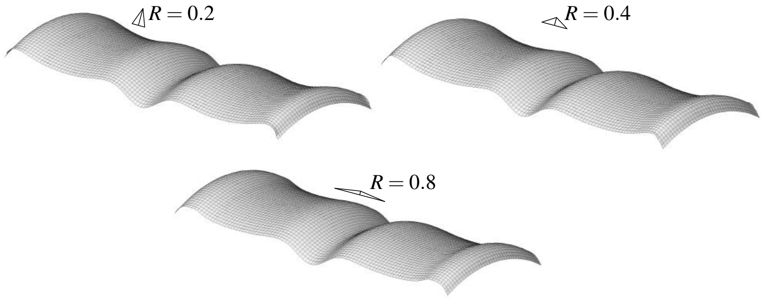


Figure 6.8: Optimization results with variable linking for different filter radii

Finally, the fitness of the designs obtained with and without variable linking are compared. Fitness is measured in terms of strain energy, such that the fittest design offers the largest value with respect to the objective function (figure 6.9). This comparison shows that the design which was optimized using variable linking stores significantly less strain energy and so offers less flexibility than the design that was obtained without variable linking. Hence it can be noted, that for this example the configuration exhibiting hinge lines is more flexible than the corrugated bellow design, which still manifests the initial curvature of the cylinder.

design	strain energy
initial design	$4.45\cdot 10^{-5}$
optimized w variable linking ($R = 0.4$)	$1.10\cdot 10^{-4}$
optimized w/o variable linking	$4.40\cdot 10^{-4}$

Figure 6.9: Comparison of strain energy

6.2.3 Spherical cupola

In the previous section, the generation of line hinges and localized corrugations was demonstrated for a cylindrical shell segment. In this example, a positively doubly curved shell is considered. The investigated structure is a cupola, fitting a rectangular footprint. The footprint is set to 10×10, and the shape of the cupola is defined by the height of the arches connecting adjacent corners of the footprint $(h_a = 3)$ and by the height of the cupola at the midspan $(h_a = 5)$. The geometry of the cupola is shown in figures 6.10 and 6.11.

The structure is supported and loaded using pre-defined displacements. Thus, each low point of the cupola is assigned a displacement $\hat{\mathbf{u}}$ in the outward diagonal direction of the structure's footprint. A displacement of 0.4 is defined in x-direction and y-direction each, so that each low point undergoes a total deformation of 0.5657. The signs of the x- and

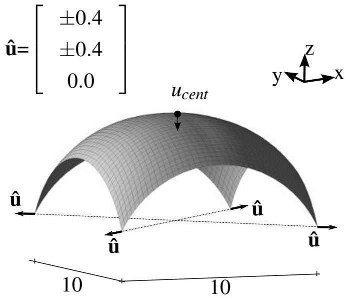


Figure 6.10: Perspective view of spherical cupola shell

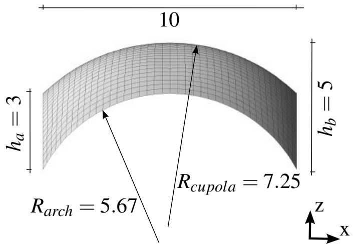


Figure 6.11: Side view of spherical cupola shell

y-components are chosen such that the deformation is oriented in outward direction with respect to the footprint. Displacements in z-direction are defined to be zero in **û**.

For this cupola, shape optimization is performed in order to maximize the vertical deflection u_{cent} of the shell's highest point. Considering yield line theory of plates, the optimal kinematic mechanism for a square plate supported at the edges would be a cross-shaped hinge line pattern, dividing the plate into four square sub-slabs. Transforming this expectation to the cupola, an optimized design showing an x-shaped pattern of regions possessing mono-axial curvature is expected. These mono-axially curved shell parts would act as line hinges, facilitating the rotations of the doubly curved shell parts.

Figure 6.12 shows the optimized shell design. The design is in good accordance with the expectations, exhibiting four mono-axially curved line hinges arranged in x-shape in the center of the shell. In Figure 6.13, the initial and the optimized design are compared for the shell's mid-section A. The section shows that the curvature is removed almost completely along the section line, starting from the center towards the free edges.

The deformations of the initial and the optimized designs are compared in figure 6.14. For the initial design, the maximum deformations occur at

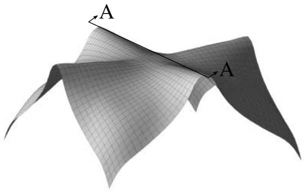


Figure 6.12: Optimized design of spherical cupola shell

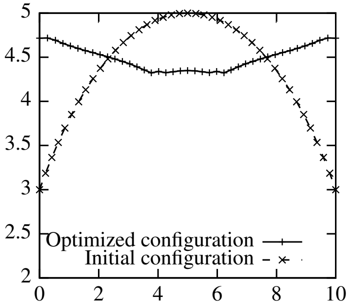


Figure 6.13: Spherical cupola shell: section A

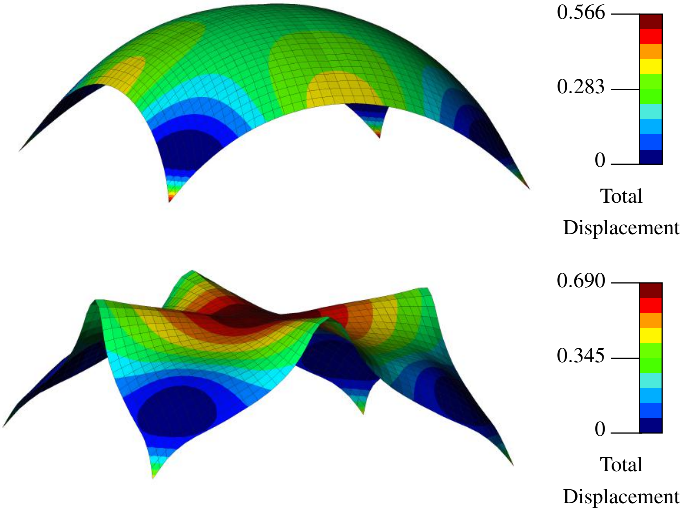


Figure 6.14: Comparison of displacements. Top: Initial design, Bottom: Optimized design

the supports where the displacement of 0.5657 is prescribed. The traced displacement u_{cent} is 0.478 for this design. In contrast, the optimized design deforms the most in the center point, where the target displacement is increased to 0.6899.

6.2.4 Normal force actuated beam

The previously discussed examples both focused on bending dominated structures. In this section, a primarily normal force effected beam is considered, which is meant to undergo a maximum vertical deflection caused by horizontal loading. The system of consideration is a beam with a length of 10, width of 1 and a varying thickness, modeled using shell elements. Support conditions are statically determined, whereat a horizontal force F_x is acting at the roller supported end of the structure. Objective of optimization is the maximization of the vertical deflection $u_{z,cent}$ in the center of the beam. Two different approaches are compared in order to initiate the vertical deflection of the initially flat beam structure. In the first approach, a tiny vertical load $F_z = 10^{-6} \cdot F_x$ is applied at the central section. In the second one, a geometrical imperfection is included into the model. For this purpose, the finite element nodes along the center line $x = 5.0$ of the system are distorted in z-direction by $\Delta z = 0.0005$. The configuration of the system is shown in figure 6.15, including both approaches for the initiation of the vertical deformation.

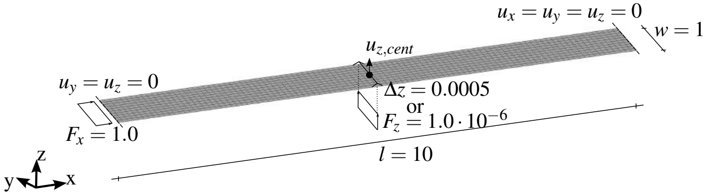


Figure 6.15: Statical system of normal force actuated beam structure

As the system's behaviour is dominated by normal force, the expected design is a kind of corrugated bellow structure, which leads to high localized bending effects. The interesting questions arising now are which shape the corrugated bellow will assume concerning its wave length and its number of waves, and which influence the chosen filter radius has onto the final design of the corrugated bellow. In order to answer these questions, optimization is performed multiple times using different shell thicknesses and filter radii. For this purpose, shell thicknesses of $t = 0.05$, $t = 0.075$ and $t = 0.1$ were used. The investigated filter radii started at $R = 0.3$, which is three times an element edge and can be considered as a minimum filter size in order to obtain results free from numerical noise (Firl [52] proposed a radius-to-edge ratio of 4 or more in order to obtain results free from discretization effects). Starting from this value, filter size is increase up to a final filter size of $R = 0.6$.

In figure 6.16, optimized designs obtained by the variant including F_z are summarized. The results show that the smallest shell thickness combined with the minimum filter radius leads to the maximum number of wrinkles. For this combination of shell thickness and filter radius, the wrinkles spread out over half of the beam length, and the remaining parts of the beam remain comparatively flat. With an increase of the shell thickness, it can be observed that the length of the complete corrugated bellow is reduced, while at the same time the un-corrugated parts of the beam get inclined. By this mean, a kind of a three-hinged frame structure is generated, with the short corrugated bellow acting as a localized central hinge. This enables a kind of rigid body deformation of the inclined un-corrugated parts of the beam. In contrast, a more global design change is preferred for thinner structures, dominated by elastic structure behaviour. Therefore it can be determined that the optimal design of the corrugated bellow is influenced by the structural boundary conditions, like the bending-stiffness-to-tensile-stiffness ratio of the structure. In order to determine the influence of different designs onto the performance of the structure, three shapes are chosen (Design A, Design B and Design C in figure 6.16) and analyzed for all three shell thicknesses. Within this

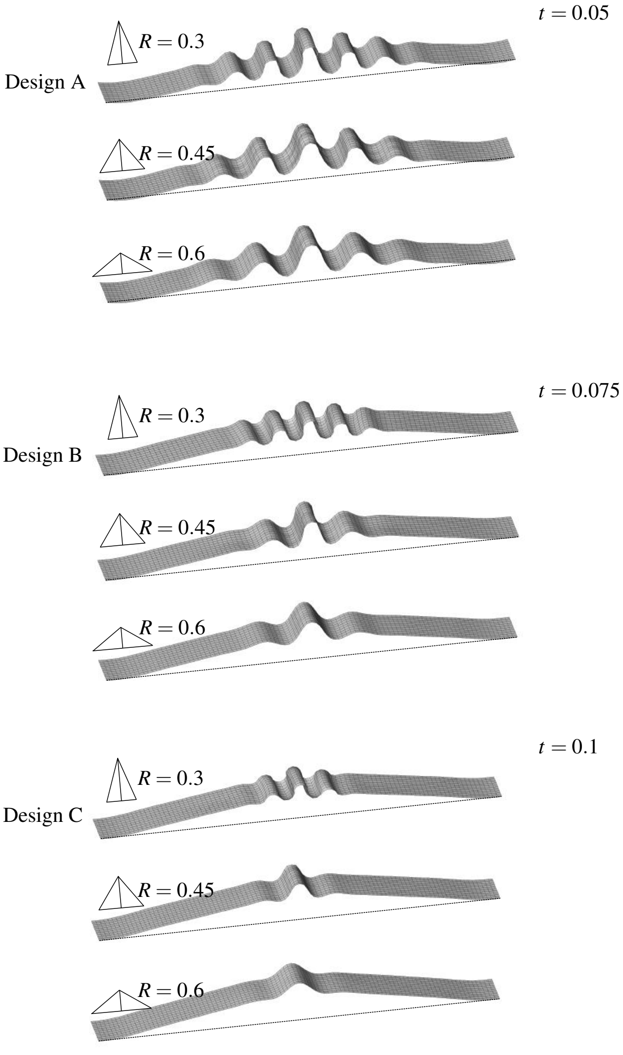


Figure 6.16: Summary of optimized corrugated bellow designs

study, also different load levels are investigated, thus the load level which was assumed for optimization ($\lambda = 1.0$) and an increased load level of $\lambda = 2.0$. The resulting displacement values $u_{z,cent}$ are summarized in figure 6.17. The presented data shows that each design performs best with respect to the deformation $u_{z,cent}$ for the shell thickness it was optimized for. Although the compared designs are quite different with respect to their shape, the differences in performance are astonishing small. Designs A and B provide comparable performance for a shell thickness of $t = 0.05$, for example, as well as Designs B and C do for $t = 0.1$. This insight conforms to results presented in Firl [52] or Firl, Wüchner, Bletzinger [54], showing that non-convex shape optimization problems may possess several local optima which lead to comparable fitness with respect to objective function values.

	$t = 0.05$	$t = 0.075$	$t = 0.1$
Design A, $\lambda = 1.0$	1.296	0.137	0.049
Design A, $\lambda = 2.0$	3.954	0.384	0.111
Design B, $\lambda = 1.0$	1.295	0.153	0.056
Design B, $\lambda = 2.0$	3.903	0.420	0.121
Design C, $\lambda = 1.0$	1.064	0.150	0.056
Design C, $\lambda = 2.0$	3.718	0.391	0.122

Figure 6.17: Comparison of deformations $u_{z,cent}$ of different designs for different shell thicknesses

The next open question concerns the influence of the filter radius. Figure 6.16 shows that the filter radius influences the wave length and the shape of the corrugated bellow, as expected. Thus an increasing filter size increases the wave length of the corrugations as well, but a direct relation, like the wave length of the corrugations depending linearly on the filter radius, cannot be observed.

Finally, the modeling techniques for initializing the vertical deformation are compared. For this purpose, we consider both designs obtained for a shell thickness of 0.05 and a filter radius of 0.3 (Design A). The designs are compared in figures 6.18 and 6.19. According to a shallow comparison, the designs seem to be identical. In order to obtain a more detailed insight, the sections B of both results are superimposed in figure 6.20. The superposition reveals that the wave lengths of both corrugated bellows are identical, the only small differences which can be observed are related to the amplitudes of the eccentric corrugations.

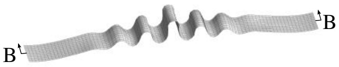


Figure 6.18: Optimized design using vertical force

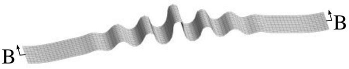


Figure 6.19: Optimized design using imperfection

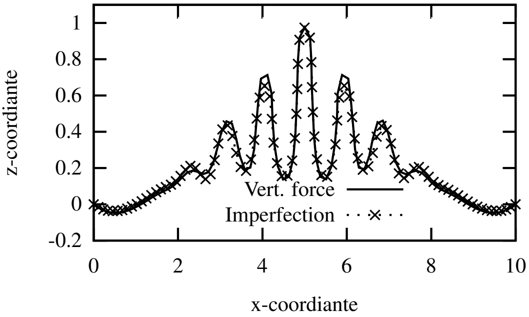


Figure 6.20: Comparison of designs using F_z and imperfection along section B

This comparison shows that the shape of the corrugated bellow is not a pure side effect of the chosen vertical disturbance. As almost identical designs are obtained for the purely normal force effected imperfect structure, the designs obtained and presented in this section can be considered to be driven by pressure force F_x.

Figure 6.21 illustrates the deformation under consideration of geometrically nonlinear effects of the corrugated structure (Design A) up to a load factor of $\lambda = 2.0$. The deformation process shows that the tensile stiffness of the thin structure still is sufficient to avoid a major horizontal contraction caused by the normal horizontal load, instead the structure performs a movement in z-direction, which leads to the intended large vertical deflection.

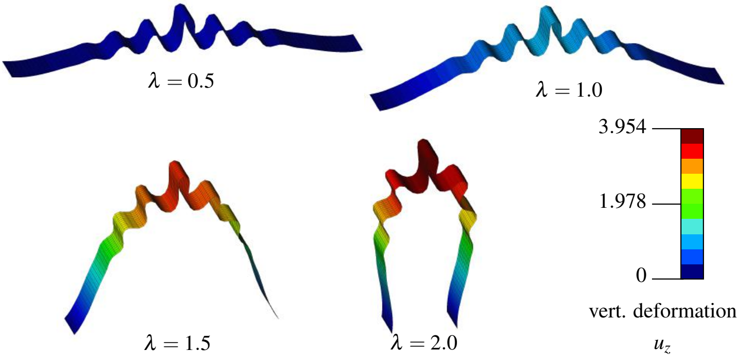


Figure 6.21: Deformation process of corrugated bellow

6.3 Conclusions

In this chapter, hingelike kinematic mechanisms for curved thin structures have been investigated. Possible mechanisms were discussed theoretically, and shape optimization techniques have been applied successfully in order to enable these mechanisms for different types of shell structures. Thereby line hinges turned out to be the preferable design for bending dominated shells. Reducing the design space in order to enforce a corrugated design led to a convex optimization problem. This resulted in a

unique corrugation pattern for different filter radii, whereat the previous line hinges were removed by single corrugations.
The last example applied optimization to a normal force dominated structure. Here multiply corrugated structures could be observed. A central conclusion which can be drawn from this example is that shape optimization of a multiply corrugated structure or a corrugated bellow is a highly non-convex problem. Hence the solution obtained by a gradient based optimization scheme corresponds to a local optimum, and the corresponding design depends on the chosen set of start parameters with respect to structure as well as to the optimization algorithm. Nevertheless, optimization results show that quite different designs may possess comparable fitness with respect to the objective function in such a highly non-convex design space. This kind of behaviour also was observed by Bletzinger, Firl, Fischer [27] for other applications using the parameter free optimization approach.

CHAPTER 7

Optimal Actuation[1]

7.1 Introduction

In chapter 6, kinematic mechanisms of shell structures have been discussed. Now these considerations are applied to actuated smart structures and the issue of how to perform actuation in an optimal manner. The basic idea of Optimal Actuation is to generate structures which provide a maximum response to an actuator input by a minimum actuation effort. For this purpose, the optimization problem introduced in equation 5.3 (section 5.3.1) is enlarged by an additional field of design variables, which is the field of actuators - therefore design variables are now defined by the shape control field s and the actuator control field a. In this context, an actuator is any tunable device able to influence the structure mechanically, such as pneumatic cylinders or piezoelectric elements. As the field of actuators was introduced, the idea of Optimal Actuation also can be used for optimal actuator placing and tuning. The final goal of Optimal Actuation is to enable a highly efficient controlling of the system, characterized by a maximized actuation response B obtained by a minimum of energy input. B can be measured in terms of strain energy or nodal displacements, for example. If we furthermore assume that the energy consumption of the

[1] Parts of this chapter have been pre-published in Masching, Bletzinger [90]

actuation process is proportional to the integral p-norm of the actuator field over the entire design space, the problem of Optimal Actuation can be described as a multi-objective optimization problem of the form

$$min\, C(a) = \int\limits_{\Omega} |a(\xi)|^p \, d\xi \tag{7.1a}$$

$$max\, B(a,s) \tag{7.1b}$$

where the function $C(a)$ indicates the actuation costs and $B(a,s)$ can be interpreted as the actuation benefit. Considering the cost-to-benefit ratio as a measure of the efficiency of the actuation, equations 7.1a and 7.1b can be summarized to the minimization of the inverse efficiency function $E(a,s)$:

$$min\, E(a,s) = \frac{C(a)}{B(a,s)} \tag{7.2}$$

7.2 An extended sensitivity filtering scheme

As Optimal Actuation introduces a new field of variables into the optimization problem, this also has an impact on the procedure of optimization. While equations 5.3 and 5.2 assumed only one field of design variables controlling the shape and eventually the sizing parameters, we now have to consider the enlarged problem of optimizing with respect to two different fields of design variables, s and a. Considering the state equation S and the field of state variables u, the final optimization problem reads

$$\begin{aligned} &min\, E(s,a,u) \\ &st. \\ &S(s,a,u) = 0 \end{aligned} \tag{7.3}$$

According to equation 5.14, the new field of design parameters a is also controlled by a gradient field p_a. As the response function f probably shows completely different behaviours with respect to design modifications in s and a, it is reasonable to link the related design updates to the gradient field via independent filter kernel functions F_s and F_a

$$\begin{aligned} \tilde{p_{s,i}} &= \frac{\sum_j F_s\left(\xi_j,\xi_i\right)\cdot p_{s,j}}{\sum_j F_s\left(\xi_j,\xi_i\right)} \\ \tilde{p_{a,i}} &= \frac{\sum_j F_a\left(\xi_j,\xi_i\right)\cdot p_{a,j}}{\sum_j F_a\left(\xi_j,\xi_i\right)} \end{aligned} \tag{7.4}$$

with $\mathbf{p_s}$ and $\mathbf{p_a}$ being the gradients of the response function with respect to the design fields for shape and actuators.

An additional argument for introducing separate filter kernels for the Optimal Actuation optimization problem is that the second field of design variables has an extreme impact on the mathematical properties of the problem. In the section 5.3, we discussed that the filter kernel cancels out for convex problem and acts as a kind of steering tool to guide the gradient based optimization algorithm towards the desired solution in case of non-convex problems. From this point of view, the filtering procedure becomes essential in applications of highly parametrized optimization problems, as the resulting design space usually is non-convex. Introducing an additional field of design variables in form of the actuator field increases this effect, especially as the field of actuators a might be non-continuous in case of a point-wise actuation. The requirement of treating both design fields adequately makes the usage of two different filter kernels essential. The following example will emphasize this necessity. We consider a statically determined tensile rod of length L with a varying tensile stiffness EA and a variable loading q. The related system is shown in figure 7.1. The distribution of the tensile stiffness and the load over the length coordinate x of the rod can be written as

$$EA\left(x,s\right) = EA_0 + \Delta EA \cdot sin\left(\frac{2\cdot\pi}{L}\cdot x + s\right)$$

$$q(x,a) = \left\{ \begin{array}{cc} 1 & x \leq a \\ 0 & x > a \end{array} \right\}$$
$$with \quad x \in [0,L]$$

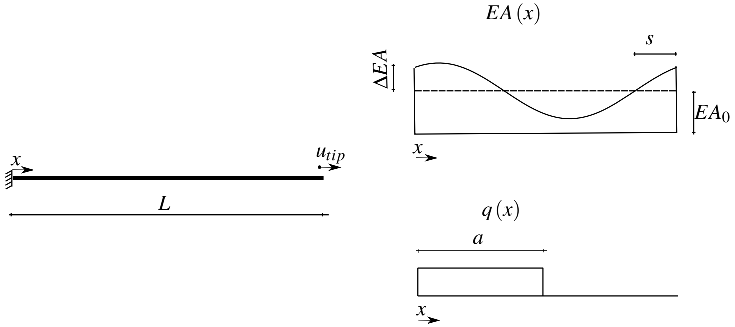


Figure 7.1: System, load and tensile stiffness distribution of tensile rod

Now we consider this rod to be optimized with respect to Optimal Actuation, where the tip displacement u_{tip} under the load q represents the actuation benefit function B and the cost function C is computed as the square of the load design variable a. The design can be modified by changing the phase change parameter s inside the tensile stiffness function, and the actuator can be effected by modifying the parameter a. The structural model and the functions $EA(x,s)$ and $q(x,a)$ were chosen such that the resulting two-parameter optimization problem can be solved analytically, but possesses the non-convex characteristics of a real-world shape optimization problem. Therefore, we come up with the following analytical formulation for the efficiency of the actuation process:

$$B(s,a) = u_{tip} = \int\limits_{x=0}^{L} \varepsilon(x)\, dx = \int\limits_{x=0}^{L} \frac{N(x)}{EA(x,s)} dx = \int\limits_{x=0}^{L} \frac{-\int_{\xi=L}^{x} q(\xi,a)\, d\xi}{EA(x,s)} dx$$

$$C(s,a) = a^2$$

$$min\ E(s,a) = \frac{C(s,a)}{B(s,a)}$$

Figure 7.2 shows a contour plot of the actuation efficiency function E in the two dimensional design space of the shape design parameter s and the actuator design parameter a. This graph clearly shows that the function behaves very differently with respect to the different types of design variables. In direction of variable s the behaviour is very smooth and represents the characteristic sinusoidal shape which was used to define the tensile stiffness $EA(x)$. In contrast, the behaviour in direction a is highly oscillating which leads to local minima of the efficiency function (figure 7.3).

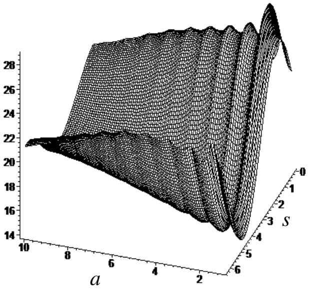


Figure 7.2: Contour plot of efficiency function in design space

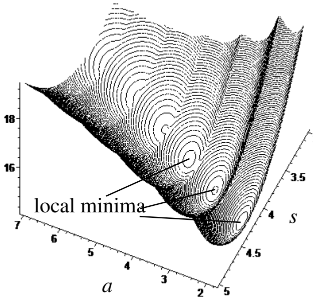


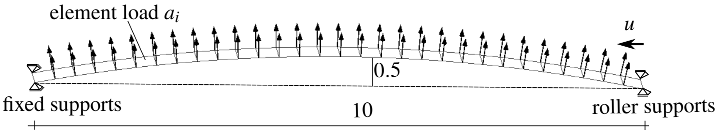
Figure 7.3: Close up of efficiency function contours

This simple example already shows that merging of different types of design variables may increase non-convexity of optimization problems significantly. In order to achieve a reasonable treatment of these kinds of optimization problems, the application of an extended filtering scheme using multiple filter kernel functions is a key feature.

7.3 Numerical examples

7.3.1 Optimization of an adaptive beam

The first example focuses on a shape adaptive beam, for which a combined shape and actuator optimization is performed. The beam has a length of 10 and an initial circular curvature with a rise of 0.5. The left edge of the system is fixed supported and the right hand side is roller supported while a pressure load is acting onto the structure according to figure 7.4. A finite element shell model is used for structural analysis and optimization. In order to keep the mechanical problem as simple as possible, geometrically linear behaviour is assumed.



Beam width	0.5
Beam thickness	0.2
Pressure load	1.0
Elastic modulus	20,000
Poisson ratio	0.0

Figure 7.4: Statical system of the adaptive beam

In order to modify the shape of the structure within optimization, all finite element nodes can be moved despite the supports, and the previously mentioned methods for in-plane regularization are used in order to avoid mesh distortion. Actuator optimization is performed by an element wise scaling of the acting pressure load. For this purpose, a design variable a_i is defined for each element, scaling the pressure acting on the specific element. The actuation benefit is measured by the horizontal displacement u of the roller support and the actuation cost is computed by summing up the acting pressures load over all finite elements, or the load controlling design variables a_i equivalently. In order to avoid arbitrarily large shape updates, an additional constraint is added to the optimization problem, which reduces the maximum allowable increase of the structure's weight to 50%. The optimization problem can be formulated in the form

$$\begin{aligned} & min \ \frac{\sum_i |a_i|}{u} \\ & st \\ & m - 1.5 \cdot m_0 \leq 0 \end{aligned} \tag{7.5}$$

with m and m_0 being the actual and the initial mass of the structure.

Two different structural optimization runs are performed for the system, one with a uniform filter kernel function for all design variables and one with individual filters for shape and actuator design variables. Filter kernel functions are used according to equation 5.7. From engineering experience, the suggested "optimal" solution would be a frame like structure with a single horizontal force acting at the roller support. Having this idea in mind, the filter kernel radii for individual filtering are chosen. In order to guide the optimizer towards the expected solution, a filter with a large filter kernel radius R_s is chosen for the shape update, while a small filter radius R_a is used in order to treat the actuator update $\left(\frac{R_s}{R_a} = 12\right)$, because of which a smooth shape update is enforced while a local actuator placement is enabled. R_a is set to 0.5, which is identical to the initial rise of the curved beam. For uniform filtering, an intermediate filter size $R = 3.0$ is used.

Figures 7.5 and 7.6 show intermediate designs during the optimization process, right at the points when the mass constraints become active.

The optimization run using individual filter kernels (figure 7.5) is striving for a solution which coincides very well with the expected frame structure. The shape already has a frame-like appearance, and the pressure load at the roller support already changed its direction, now pointing in

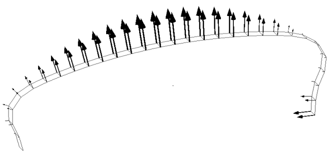


Figure 7.5: Intermediate design using individual filter kernels

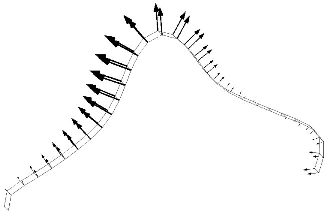


Figure 7.6: Intermediate design using uniform filter kernel

direction of the investigated displacement u. At the same time, the uniform filter approach shows a completely different design with a peak in the middle of the structure. By this mean, a large lever arm is generated, leading to large bending deformations, also increasing the efficiency of the actuation process. The pressure load distribution looks similar to the individual filtering case, as the change of sign close to the roller support is also visible here. These features are kept upright until the converged solution of the optimization is reached (figures 7.7 and 7.8). In case of

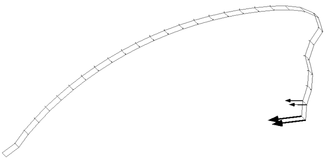


Figure 7.7: Optimization result using individual filter kernels (250 iterations)

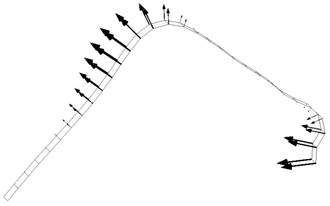


Figure 7.8: Optimization result using uniform filter kernel (170 iterations)

the individual filtering, a frame structure with a vertical and a diagonal element has developed. The actuating pressure load is concentrated at the roller support, as it was expected, while the pressure acting onto the rest of the structure has been optimized to zero. The vertical member of the structure is designed in a corrugated form, which leads to a combined load carrying behaviour by bending and normal force, which again increases the flexibility of this member. As the wave length of the corrugations is much smaller than the filter radius applied to shape design variables, it can be concluded that the shape of the corrugations is governed by the geometry and stiffness of the structure, and is not a direct consequence of the filtering procedure.

In contrast, the uniform filter approach still shows the characteristic peak, and the load is not concentrated, but distributed over the peak and the support region. Although this design is very different from what was expected beforehand, it is also a quite efficient way of actuation because of the large lever arm generated by the peak. Due to the uniform filter size for both fields of design parameters, the optimizer is not able to generate a highly localized actuation as we can see it in the case of the individual filtering (figure 7.7).

A comparison of objective function values reveals that both designs (figures 7.7 and 7.8) show a quite comparable performance with respect to the objective function. So the design obtained by individual filtering leads to a final objective value of 1.9% times the initial objective value f_0, where the uniform filtering ends up with an objective of $2.5\% \cdot f_0$.

In order to complete the comparison, a last optimization is performed, using individual filter kernels with an inverted size ratio $\left(\frac{R_s}{R_a} = 0.2\right)$. Figure 7.9 presents the final design, and the similarity between this design and the one obtained by a uniform filter kernel (figure 7.8) is obvious. This confirms the statement made in section concerning shape control, where it is claimed that the filter cancels out. So different filters might lead to identical or very similar designs, if optimization is directed into the radius of convergence of the same optimum in the early optimization phase.

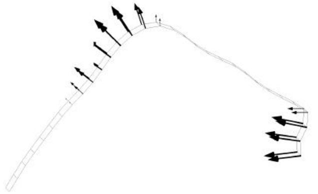


Figure 7.9: Optimization result using individual filter kernels with inverted size ratio $\left(\frac{R_s}{R_a} = 0.2\right)$

7.3.2 Generation of complex bead designs

This second example is meant to show the capability of the method to generate complex flexible structures which can act as elastic hinges providing high movability of the structure. These hinges usually are implemented as corrugated bellows, and as the direction of load application as well as the applied support conditions are usually not known beforehand, rotational symmetric bellows are used most frequently, in order to be applicable to a multitude of applications. The following example will focus on the generation of custom tailored elastic hinge structures for specific geometrical and structural boundary conditions by applying methods of parameter free shape optimization.

From a methodical point of view, this example is also very interesting, as it presents a geometry which cannot be optimized by pure intuition anymore. Additionally it emphasizes the benefit of the node based optimization approach towards CAD based optimization, as very complex geometries can be realized. Setting up a CAD based optimization model being able to reflect a complex and short waved bead pattern on the surface of the cylinder would be time consuming, especially when orientation, width and length of the beads are unknown. The chosen CAD parametrization would have significant influence on the final optimal design anyway. The parameter free optimization approach supersedes this modeling step, as well as re-meshing or mapping operations between design and computational model.

The system to be considered is a cylinder of length $l = 30,000$ and of diameter $d = 10,000$. One end of the cylinder is beveled in 45 degrees, and all translatoric degrees of freedom are fixed at this edge (see figure 7.10).

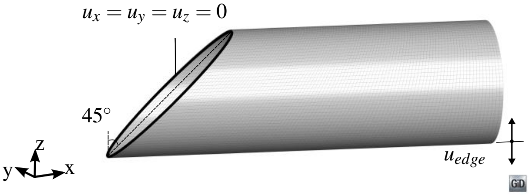


Figure 7.10: Support conditions of beveled cylinder

Initially, the structure is loaded by a vertical load which is uniformly distributed over the entire body of the cylinder and node-wise controlled via the actuator variables **a**. In the context of an optimization with respect to Optimal Actuation, the sum of all acting external forces is considered as the cost function, while the vertical deflection of the un-beveled edge u_{edge} of the cylinder is used to measure the benefit of the actuation. Analogously to the previous example, this structure is to be optimized with respect to Optimal Actuation by modifying the shape of the cylinder as well as the placing of the acting external forces. Therefore, the cost-benefit optimization can be formulated as

$$min \frac{\sum_i |a_i|}{u_{edge}} \tag{7.6}$$

The results of this combined optimization are shown in figure 7.11. Again an individual filtering for shape optimization and actuator placement has been used. Gradients with respect to shape are treated using a filter radius $R_s = 1,000$ in order to ensure sufficiently large wave length and curvature

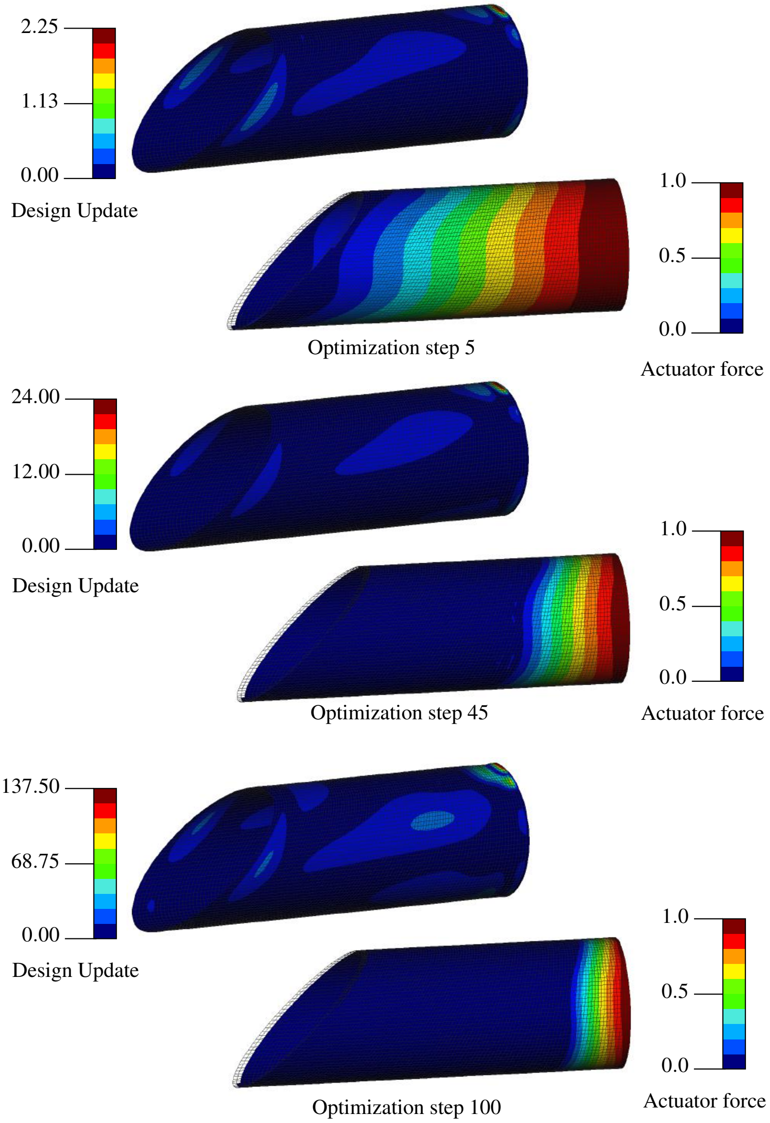


Figure 7.11: Combined shape optimization and actuator placing

of design features, while actuator placement is modified using a smaller filter radius of $R_a = 200$. The results clearly show that actuator placement is dominant in this case, while shape modifications are tiny. Optimization is striving for a solution only actuating the free edge of the cylinder, as this loading provides the largest lever arm and correspondingly the highest actuation benefit.

Based on these results and in order to accelerate computations, an actuating force p_z only acting on the un-beveled edge of the cylinder is assumed for the ongoing optimizations, which deal with pure shape optimization of the cylinder. Within this optimization, two different load cases will be considered. For this purpose, the vertical load acting at the edge of the cylinder can be applied in upward or downward direction, as indicated in figure 7.12.

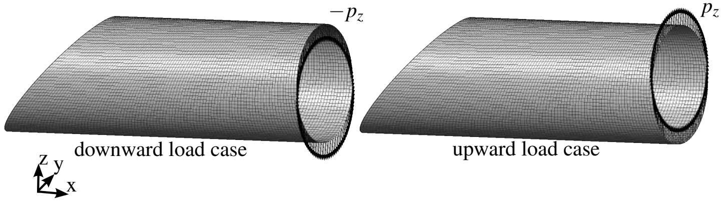


Figure 7.12: Different load cases of beveled cylinder for pure shape optimization

As actuator placement was already performed and therefore the cost function remains constant, the objective of the optimization becomes the maximization of the vertical deflection at the loaded edge u_{edge} induced by the ring load. Three different optimization runs will be performed reflecting the possible combinations of load cases. During the first two optimization runs, only one loading direction is considered and the related displacement is maximized. In the last optimization run, both tasks are combined in a multi-objective optimization, where each load case is considered separately and the related displacement responses are optimized as an equally weighted sum. In order to obtain a bead pattern instead of a free shape optimization, the design space is limited. Each node is allowed to move ±250 in out-of-plane direction.

Optimizations are performed under consideration of large deflections. Therefore, fully geometrically nonlinear analyses are performed, and sensitivities are computed based on the nonlinear equilibrium condition. For this purpose, sensitivity formulations according to A.2 are used.

Optimization results shown in figures 7.13 and 7.14 reveal that no classical corrugated bellows can be identified, due to the prescribed loading directions and the inclined supported edge. All three designs have in

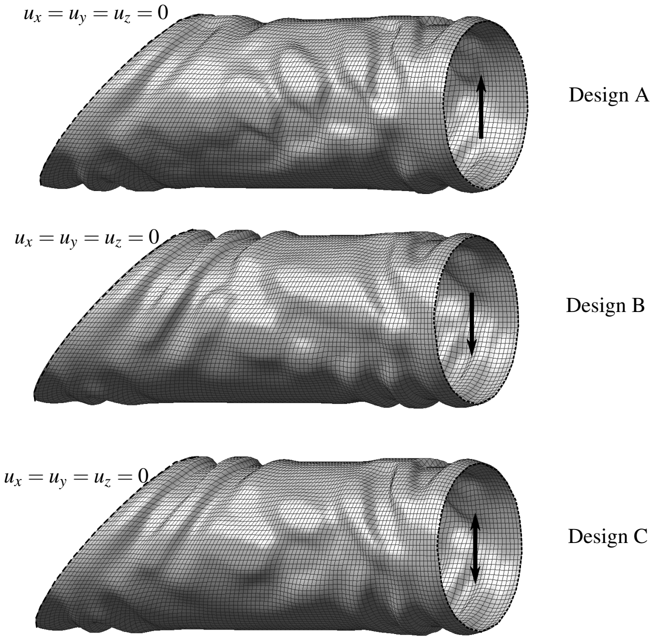


Figure 7.13: Iso parametric views of optimal designs (A: Load case upwards, B: Load case downwards, C: Both load cases)

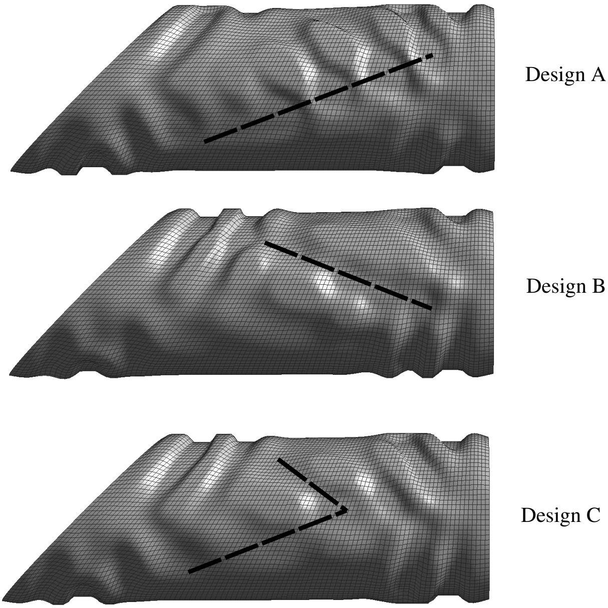


Figure 7.14: Side views of optimal designs (Top: Load case upwards, Center: Load case downwards, Bottom: Both load cases), dashed lines indicate main beads on the lateral surface of the cylinder

common the corrugated bellows at the edges of the cylinder, located in the tension and compression zones. Strong differences can be seen on the lateral surface of the cylinders. Designs A and B of figure 7.14 are dominated by diagonal main beads on the lateral surface, whereat the orientation results from the loading direction. These beads are acting as a kind of tension member, transporting the load over the length of the lateral surface, whereas the rest of the cylinder body performs an almost stress free deformation.

The multi objective optimization, which was formulated as an equally weighted sum of the response functions of the two other optimization runs, tries to find a compromise out of the two designs described above, as expected. As a consequence, no distinct diagonal bead exists anymore. A slight Y-shaped deepening can be observed only in the left part of the cylinder. The corrugated bellows at the loaded edge are almost symmetric, which is not surprising due to the altering load directions and the long distance to the inclined supported edge.

In order to judge the influence of the filter size on the final result of the optimization, a parameter study is performed for the upward load case. For this purpose, the filter radius is reduced step-by-step from the initial size of $R = 1,500$ to a final size of $R = 375$. Figure 7.15 compares the obtained optimization results, where the contour plot highlights the directed design update, red color indicating an outward design update while the blue color indicates an inward design update. It can be observed that the chosen filter size has an influence on the final design, which is an expected behaviour for a complex shape optimization problem, being definitely non-convex. Despite these differences in the final shape, basic features can be identified, which are characteristic for all four compared shapes. In all cases, outward oriented beads can be observed in the central part of the lateral surface of the cylinder, oriented diagonally from top-left to bottom-right. V-shaped beads of changing orientation developed close to the supported edge, being orientated top-right to bottom-left on the top, and top-left to bottom-right on the bottom part of the cylinder. These beads start do widen with an increasing filter size, as expected, and tend to unite. Therefore, the beaded areas on the cylinder's lateral surface become larger when the radius of the filter is increased.

In summary, it can be noted that the filter influences the design in the expected way, such that a larger filter emphasizes an optimal solution showing overall shape changes, while a smaller filter leads to localized design modifications. Nevertheless, all designs are clearly governed by the pure gradient information of the response function, leading to basic

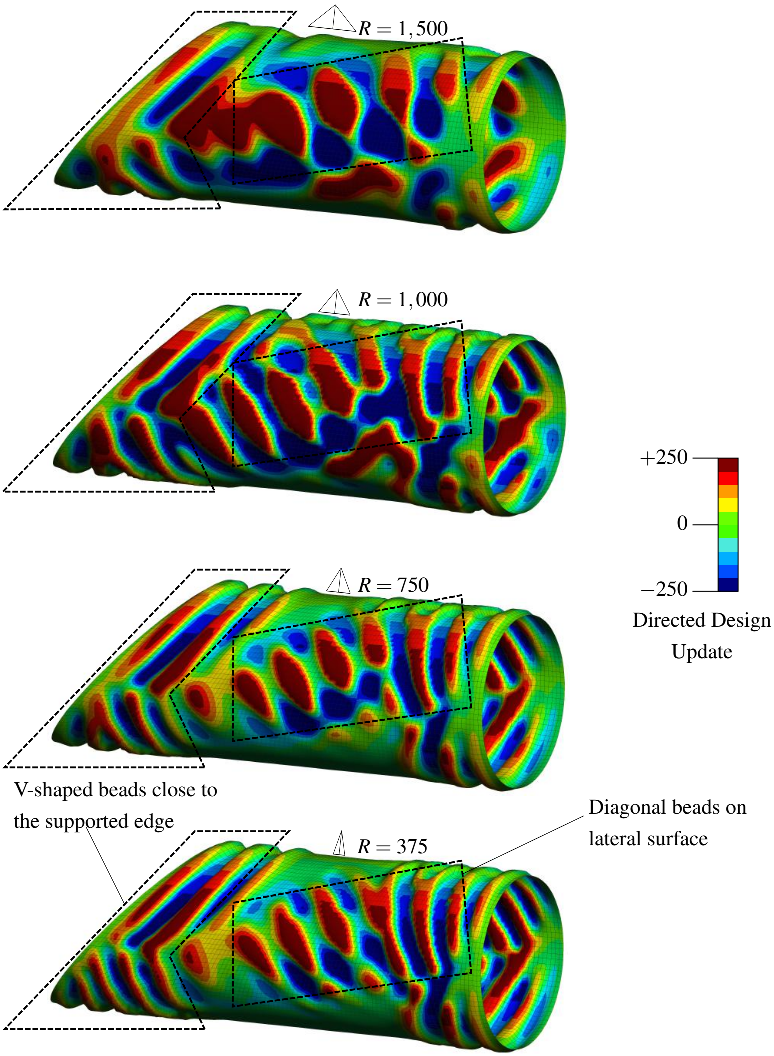

$R = 1,500$
$R = 1,000$
$R = 750$
$R = 375$
+250
0
−250
Directed Design Update
V-shaped beads close to the supported edge
Diagonal beads on lateral surface

Figure 7.15: Upward load case, optimized using different filter sizes

characteristic elements in the design. These elements are preserved and remain present, even for severely differing filter sizes.

In figures 7.16 and 7.17, the structures which were optimized using one loading direction are shown in different states of deformation. In both cases, a maximum deformation of $10,000$ is applied, which is identical to the cylinder's diameter. It can be determined that the bead structures act in the expected way. The structure shows a crinkle-like behaviour along the lateral faces of the cylinder, which is well known from membrane structures effected by compressive strains. As this kind of local crinkling is a nonlinear process and the observed displacements are significant,

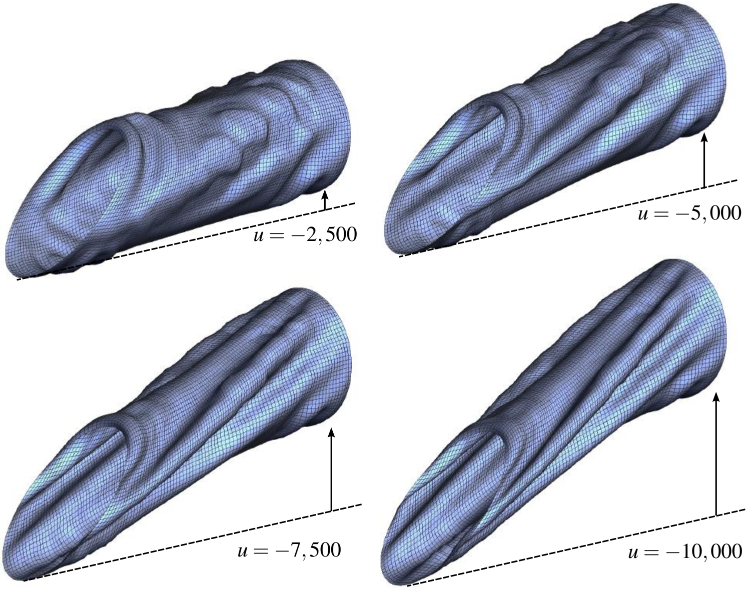


Figure 7.16: Deformation progress of Design A under loading in upward direction

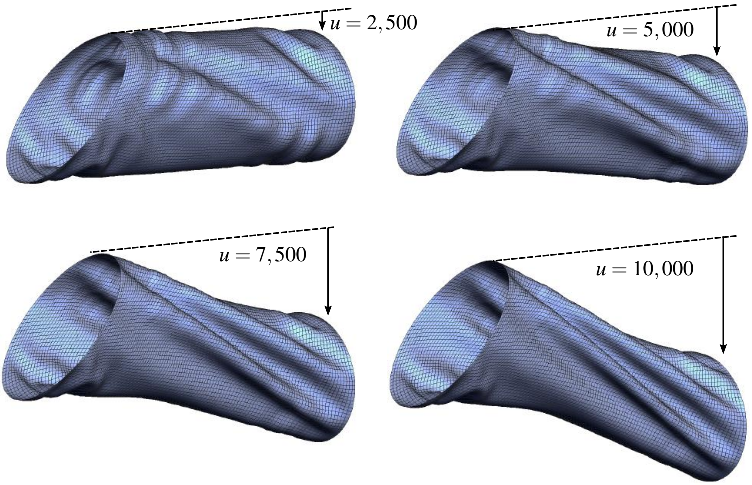


Figure 7.17: Deformation progress of Design B under loading in downward direction

the consideration of geometric nonlinear effects and large deformations within the structural analysis as well as within the optimization is essential.

In figure 7.18, the load-displacement curves for the different cylinder designs A,B and C are compared. As recommended by Choong, Ramm [41], dynamic effects have been taken into account in the analysis in order to ensure numerical stability of the computations. In pure static analysis, the initial design and Design C fail for moderate deformations due to local instabilities. In order to overcome these instabilities, a dynamic analysis using a Newmark-Beta integration scheme was performed, using a density of $\rho = 7.5 \cdot 10^{-7}$ and stiffness proportional Rayleigh damping. Density has been chosen as small as possible, such that computations are stable but the structure does not show observable dynamic effects. It can be deter-

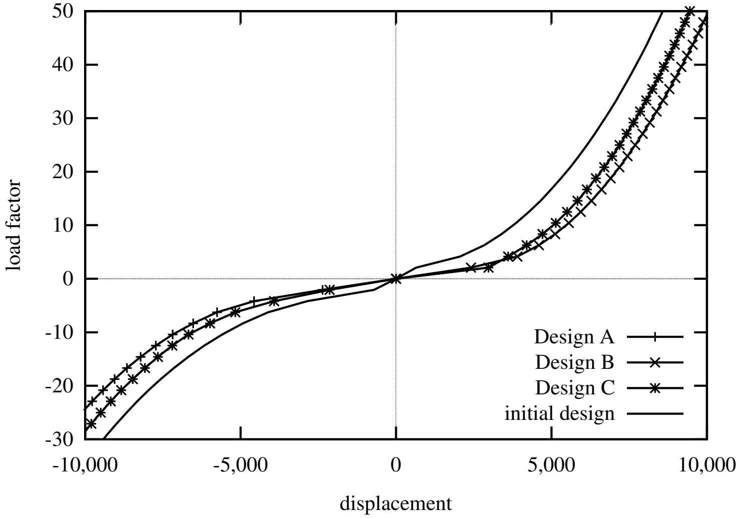


Figure 7.18: Load-displacement curves of different cylinder designs

mined that the individually optimized designs A and B show the softest behaviour and so the best adaptivity, as expected.

Design C shows a surprisingly high flexibility, being quite comparable to the designs A and B, especially for deformations ranging from $-5,000$ to $+5,000$. However, in large displacement regimes, static computations collapsed for both loading directions, which indicates sudden instability failure and a non-robust behaviour of the structure for large deflections. In summary, the compromise design C shows a good flexibility for both loading directions, but it does not behave as resistantly as the designs tailored for a specific loading direction.

As a conclusion of this example, it can be noted that the optimization approach based on pure engineering experience fails for quite simple geometries already. Methods of structural optimization have to be taken into account, especially for those optimization problems facing tasks of large deflections. Beyond that, this example points out the advantages of the finite element based structural optimization approach, not being based on an a-priori choice of geometry variables. This choice naturally has an influence on the final optimization result. In contrast, the finite element based optimization approach allows optimization with a minimum of modeling effort also for changing boundary conditions, such as multiple or varying loads.

7.3.3 Shape adaptive wing

In this section, the presented methods are applied to a real structure. As an example, an adaptive wing is used, which currently is object of research under the lead of Onur Bilgen [22] [23] at the Old Dominion University and at the Swansea University. The investigated wing belongs to an unmanned light-weight aircraft, which is equipped with piezoelectric elements instead of classical flaps, adapting the wing by selective form modifications to the current flight and air-flow situation. Figure 7.19 shows a schematic drawing of the wing, including the three piezoelectric actuators in red colour. The underlying airfoil corresponds to a NACA 0012 profile.

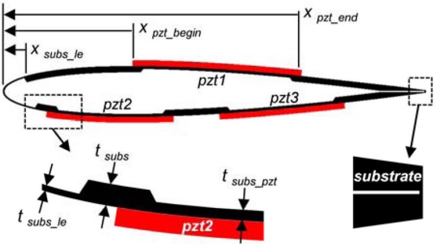


Figure 7.19: Schematic drawing of the adaptive wing [23]

The works in Swansea and Norfolk mainly focused on the experimental determination of optimal airfoil geometries in order to obtain maximum lift-to-drag ratios. Structural analysis was performed using 2D-models, actuator placement and choice of substrate thickness was done using genetic algorithms. In line with this thesis, the wing is to be optimized with respect to shape adaptivity using gradient based optimization applying the ideas of Optimal Actuation. For this purpose, both actuation load case and fluid load case are considered. Numerical fluid load computation is based on the open source airfoil analysis tool XFOIL [50], which also was used in Swansea. Objective of optimization is to reach a pre-defined shape by actuating the wing with a minimum actuation effort, at the same time deformation limits may not be violated in the un-actuated configuration with only the fluid load acting.

The most extreme operational state of the wing presented by Bilgen (see figure 7.20) is chosen as target geometry for the actuation process.

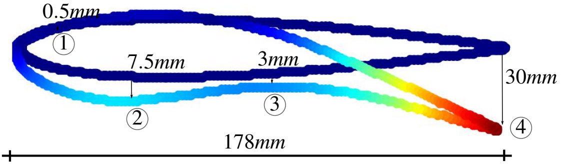


Figure 7.20: Traget deformation for actuation (based on [23])

Throughout the optimization, this shape is described and and evaluated via the four illustrated nodal displacements. For analysis and optimization purpose, an extruded version of the 2D-model used in Swansea with a constant chord length is used (figure 7.21). Temperature loads are used to mimic the effects of the piezo elements. Due to the extrusion, a three dimensional model of the wing is generated, which is meshed using eight-noded hexahedral elements.

A first optimization is performed in order to obtain reference values for the temperatures in the actuators which are necessary in order to actuate the initial wing design. In this optimization, the shape of the wing remains untouched, while the temperature loads **a** acting in the actuators are opti-

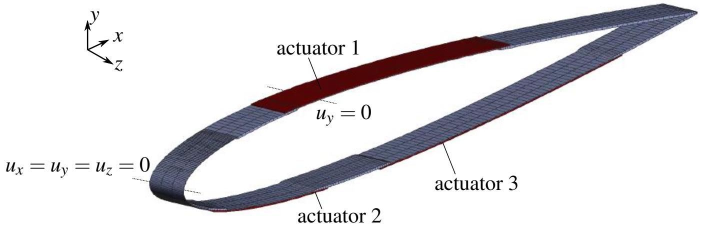


Figure 7.21: Finite element model of adaptive wing with actuators and support conditions

mized. A least-squares sum of the traced nodal displacements marked in figure 7.20 is used as response function (see equation 7.7).

$$min\ f(\mathbf{a}) = \sum_{i=1}^{4} \left(u_i(\mathbf{a}) - u_i^{ref} \right)^2 \qquad \begin{array}{l} u_1^{ref} = 0.5mm, \quad u_2^{ref} = 7.5mm \\ u_3^{ref} = 3mm, \quad u_4^{ref} = 30mm \end{array} \tag{7.7}$$

Obtained load values and the corresponding characteristic displacement values are listen in figures 7.22 and 7.23. As four displacements have to be fitted, but only three design variables are available, the target values cannot be satisfied exactly. Figure 7.24 shows the wing in actuated configuration according to the Carat-simulation.

Actuator	normalized temperature
1	$a_1 = 23$
2	$a_2 = 100$
3	$a_3 = -84$

Figure 7.22: Actuator forces **a** for initial geometry

Node	Displacement
1	0.49mm
2	7.9mm
3	3.1mm
4	32.4mm

Figure 7.23: Characteristic displacements of actuated initial geometry

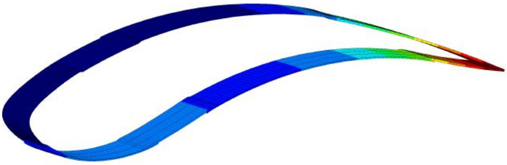


Figure 7.24: Wing in actuated shape

These load values are now used as initial values for shape and actuator optimization. In order to maintain the flow conditions around the wing, only the inner face of the wing is used as design space. Shape controlling design variables are collected in vector **s**. Objective of the optimization now is to minimize the actuation effort, which is measured via the l_1-norm of the vector **a**. Simultaneously, constraint functions are defined, forcing the wing into the desired shape under actuation load (g_1), and limiting the deformation under fluid load (g_2), whereat the investigated displacement is evaluated at the trailing edge of the wing. The considered fluid load corresponds to flow conditions under an angle of attack (AOA) of 7° and a flow rate of $30\frac{m}{s}$. Figure 7.25 shows the corresponding distribution of c_p-values according to an XFOIL computation and the resulting consistent nodal forces acting on the finite element model.

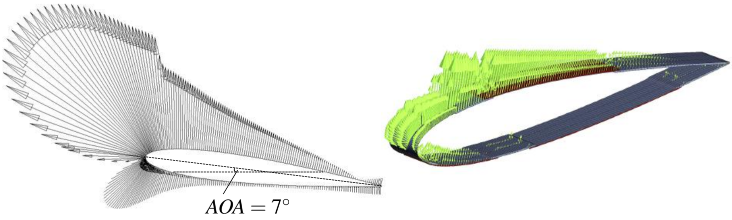


Figure 7.25: Fluid load: c_p-value distribution and corresponding nodal forces

The entire optimization problem is formulated in equation 7.8. Thereby **u** indicates the displacement in the actuated state and **v** names the displacement due to the fluid load, while indices related to displacement vectors again refer to the node numbering introduced in figure 7.20.

$$
\begin{aligned}
&min\ f(\mathbf{a}) = \sum_{i=1}^{3} |a_i| \\
&st. \\
&g_1(\mathbf{a},\mathbf{s}) = \sum_{i=1}^{4} \left(u_i - u_i^{ref}\right)^2 < TOL \\
&g_2(\mathbf{a},\mathbf{s}) = v_4 < 1.27mm
\end{aligned}
\tag{7.8}
$$

The maximum allowable deformation of the trailing edge by $1.27mm$ under fluid load is the same deformation which the initial design showed for the given flow conditions.

In this application, a filter radius of $R = 15mm$ was used for the treatment of shape determining design variables **s**. As the actuators are discontinuously distributed over the wing, a filtering of the gradients related to variables **a** is abdicated.

Design changes performed on the wing during optimization are illustrated in figure 7.26. In order to emphasize the areas of increasing and decreasing substrate thicknesses, the vector plot is scaled by a factor of 20. The most shape modifications are performed at the bottom part of the wing, where the substrate thickness was increased in the regions topping the actuators, while the material connecting the actuators 2 and 3 was thinned out. As the fluid pressure load acting in this region are very small, the resulting decrease of stiffness is not problematic.

Figure 7.27 allows a more detailed evaluation of the shape modifications. The passive material connecting actuator 2 and 3 is thinned to a lower allowable limit of $0.025mm$, in order to facilitate the change of curvature which is characteristic for the actuated wing configuration. At the same

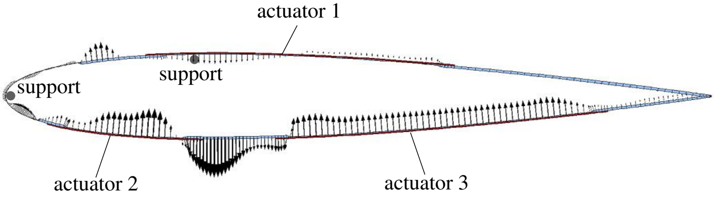


Figure 7.26: Design changes of wing's inner face (scaled by a factor of 20)

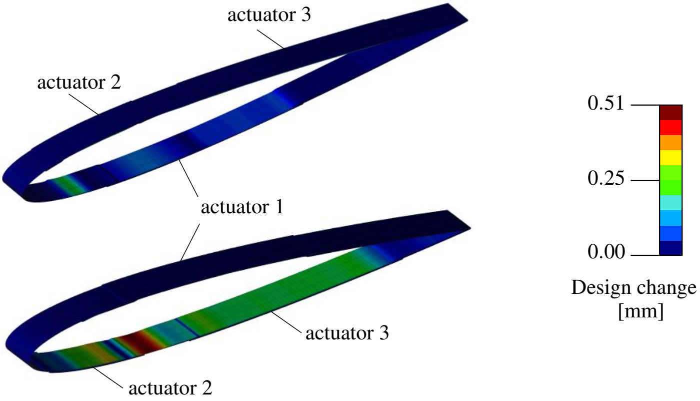


Figure 7.27: Design changes of wing's inner face. Bottom: Standard view, Top: Inverted view

time, the material layer covering actuator 3 is getting thicker, whereby the jump in material thickness between the passive material and actuator 3 is removed by a homogeneous transition. As the top part of the wing has to curve less than the lower part during the actuation process, shape modifications are smaller in this area. Here the material thickness of the substrate topping the actuator is also increased, while material is taken away at the pure passive structure.

Now the resulting actuation effort is considered. In figure 7.28, the actuation temperatures for the initial and the optimized design are compared. It can be seen that especially the actuators 1 and 2 offered a lot of potential for optimization, and up to 66% of actuation energy have been saved. For actuator 3, only a saving of 10% was achieved. As the contraction of this actuator is the main driving force in order to obtain the S-bent shape in the actuated state, a relatively high actuation force within this actuator is necessary.

Actuator	temperature load initial design	temperature load optimized design	saving
1	$a_1 = 23$	$a_1 = 9$	60%
2	$a_2 = 100$	$a_2 = 34$	66%
3	$a_3 = -84$	$a_3 = -76$	10%

Figure 7.28: Actuator loads for optimized geometry in comparison with initial design

7.4 Conclusion

In this chapter, the necessity of a modified or extended shape control algorithm for optimization including actuator placement was presented. A numerical example showed that an aspired localized actuation and a smooth optimized shape cannot be achieved by a uniform filter kernel function.

Furthermore the influence of the filter size on the shape of complex bead patterns was discussed. Here different filter radii led to different optimal designs, proofing the optimization problem to be non-convex. The filter effected the final design in the expected way, such that a larger filter emphasized an overall optimized design, while a smaller filter led to localized design modifications, whereat the characteristic design elements have been preserved.

The example considering the adaptive wing combined a continuous shape design field with discretely distributed design variables. Here also a second load case was considered, subjecting the structure in the un-actuated state and so defining a minimum stiffness of the structure. This complex optimization problem was solved successfully, leading to a significant decrease of required actuation force.

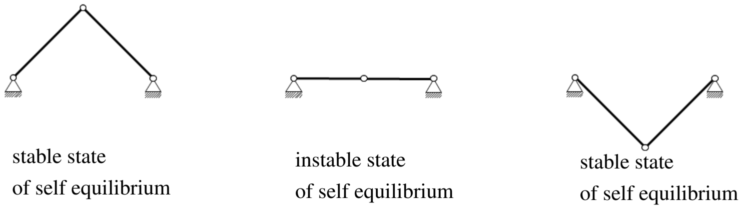
CHAPTER 8

Bi- and multistable structures

8.1 Motivation

In the chapters 6 and 7, "classical" smart structures have been considered. These structures can be actuated by a permanent actuator input. As soon as the actuator is shut down, the structure returns to its initial configuration. This necessity of permanent actuator input brings up the idea of applying bistable structures in the field of smart structures, which will be discussed in this chapter.

A bi- or multistable structure is a structure which can assume two or even more different shapes of self equilibrium without any load acting onto the structure. A very simple example of a bistable structure is the two-bar-truss structure, which is discussed quite frequently in introductive lectures to nonlinear finite elements, as it shows a distinct nonlinear load carrying behaviour despite its simplicity. Figure 8.1 shows the states of self equilibrium of the two bar truss. The structure even possesses three configurations of self equilibrium, whereof two are stable states and the third one is a instable state, which means that any disturbance applied to the position of the center node will cause the structure to a sudden snap through towards one of the stable states.



stable state
of self equilibrium

instable state
of self equilibrium

stable state
of self equilibrium

Figure 8.1: Two bar truss as example for bistability

Bistable structures show sudden changes in deformation when a critical load is exceeded. In fact, this is a buckling phenomenon, which usually is an undesired behaviour in structural design. For bistable structures on the other hand, buckling loads are excessed by intend, in order to switch between self stable configurations.

8.2 Bistable structures in morphing applications

Bistability is a very interesting effect for applications in the context of smart structures or morphing applications, and the idea of exploiting bistable properties for compliant mechanisms goes back to the 1980 [75]. For a bistable smart structure, actuation is only necessary in order to switch between configurations, while the desired shape is preserved as a state of self equilibrium. Thus, energy consumption is reduced to a minimum, as keeping the configuration upright does not require any actuator input. This property makes bistable structures also very attractive with respect to the aspect of reliability and safeguarding against failure. A "classically" actuated structure, needing permanent power supply, would immediately lose its controlled character if the energy supply was interrupted. This might lead to a sudden and complete collapse of the entire structure. On the other hand, a bistable structure only loses its ability to switch between configurations, but the load bearing capacity and the stability of the structure would not be affected immediately by a power breakdown.

8.3 State of research

In section 8.1, the effect of bistability was discussed regarding a truss structure. Generating bistability for curved shell structures is way more challenging. For this purpose, analytical formulas are used in order to achieve bistable behaviour for shells with constant curvature radii. Cylindrical shells are known to be monostable if they are made from isotropic material without any prestress [65], thus actual research mainly focuses on the generation of bistable cylindrical shells using orthotropic composite material in combination with prestress [81].

In this context, necessary prestress is very often generated by using the direction dependent thermal expansion of composites materials in the autoclave method in order to obtain prestressed cylindrical shells made from staggered composite material [93] [104] [105]. Another approach is to generate prestress by bonding mechanically prestressed rubber material as top and bottom layers onto elastic core material. By this mean, a bistable structure can be generated which switches between a cylindrical and a saddle shape [38].

The investigation of bistability for non-prestressed structures is quite a new field of research. In Coburn et al [43], a tristable positively doubly curved shell is presented for which the multistable properties are tailored by choosing adequate radii of curvature.

8.4 Finite Element based optimization with respect to bistability

8.4.1 Formulating analysis and optimization problem

Summarizing section 8.3, it can be noted that the actual design process focusing on bistability addresses structures possessing globally constant properties concerning curvature and prestress. This allows to judge about the existence of multiple stable configurations by using analytical models. Goal of this chapter is to apply finite element based structural optimization to the field of bi- and multistable structures, in order to generate free-form structures possessing several self stable configurations. For this purpose, shape optimization as well as composite fiber angle and layer thickness optimization techniques will be used.

The subsequent question is, how to set up an optimization problem aiming on the generation of a bi- or multistable structure. Structural analysis of the underlying structure is formulated as a geometrically nonlinear static problem, as the structure of consideration is supposed to show significant buckling behaviour. Accordingly, requirements with respect to bistability have to be formulated in context of the nonlinear analysis. In order to be able to specify a characteristic deformation of the system which is intended to be a self stable configuration in the optimized state, structural analysis is performed using a displacement controlled path following technique. This enables the user to specify a characteristic degree of freedom of the system together with a target displacement values, defining the aspired stable configuration. Figure 8.2 shows an exemplary load-displacement diagram, where the initial design does not posses bi- or multistable characteristics. The degree of freedom u_c, which is controlled by the path following method, is plotted on the horizontal axis, while the vertical axis refers to the load factor λ. Two reference values $\bar{u}^1$ and $\bar{u}^2$ are specified for the controlled displacement, specifying the desired self stable configurations. The load-displacement path changes as optimiza-

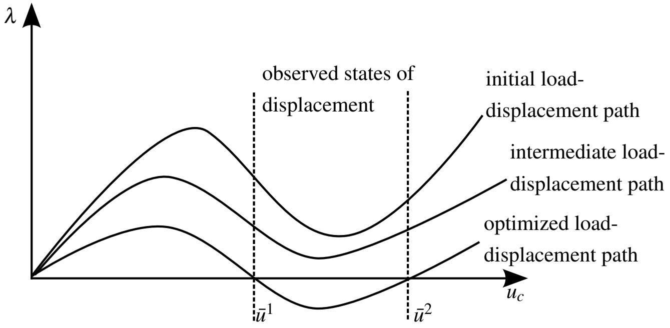


Figure 8.2: Exemplary load displacement diagrams for optimization with respect to bistability

tion proceeds, and approaches towards the final solution, where $u_c = \bar{u}^1$ and $u_c = \bar{u}^2$ define self stable configurations, such that the newly generated structure now possesses three self stable configurations, including the origin of the coordinate system $u_c = 0, \lambda = 0$.

8.4.2 Prediction in context of optimization using nonlinear analysis

In nonlinear static analysis, a Newton-Raphson method (see section 4.2.2.1) is used in order to determine the load factor λ and the displacement field $\mathbf{u}$ which satisfy the nonlinear equilibrium condition [17]. In order to speed up convergence of this iterative method, usually a predictor is used in displacement controlled analysis, providing an initial guess of the incremental displacement field $\mathbf{u}_{inc}$ and the incremental load factor λ_{inc} which are expected for the upcoming load step. This approach of applying a predictor to a Newton-Raphson method is usually referred to as a predictor-corrector approach [135].

In order to compute the predictor, the load-displacement history of the previous load step is used. For this purpose, the incremental displacement of the last load step is linearly extrapolated. Applying this idea to the controlled degree of freedom u_c, the linear extrapolation factor s can be computed based on the current displacement value $\bar{u}^i$, the displacement increment computed in the last load step $u^i_{c,inc}$ and the upcoming prescribed displacement value $\bar{u}^{i+1}$ in the following form:

$$\begin{aligned}
&\bar{u}^i + u^{i+1}_{c,inc} = \bar{u}^{i+1} \\
&\bar{u}^i + \overbrace{\underbrace{s \cdot \left(\bar{u}^i - \bar{u}^{i-1}\right)}_{u^i_{c,inc}}}^{u^{i+1}_{c,inc}} = \bar{u}^{i+1} \\
&s = \frac{\bar{u}^{i+1} - \bar{u}^i}{\bar{u}^i - \bar{u}^{i-1}}
\end{aligned} \tag{8.1}$$

The scalar factor s, which is obtained by investigating the single degree of freedom u_c, now is applied to the complete displacement field and to the load factor. Accordingly the estimated incremental displacements and the estimated load factor increment are computed by

$$\begin{aligned}
\mathbf{u}^{i+1}_{inc} &= s \cdot \left(\mathbf{u}^i - \mathbf{u}^{i-1}\right) \\
\lambda^{i+1}_{inc} &= s \cdot \left(\lambda^i - \lambda^{i-1}\right)
\end{aligned} \tag{8.2}$$

Figure 8.3 shows the prediction of point $i+1$ based on the previously computed equilibrium points i and $i-1$.

In the context of an displacement controlled optimization as illustrated in figure 8.2, the predictor procedure has to be modified. During optimization, the controlled displacement u_c is constant, but the design is changed which necessitates re-solving of the mechanical problem. As there is no displacement increment $u_{c,inc}$ in between optimization steps, the computation of the scalar factor s according to equation 8.1 is not applicable. In order to be able to generate a predictor for solving the structural problem in each optimization step or pseudo load step $i+1$ nevertheless, a modified predictor method has to be used. In equation 8.1, the driving

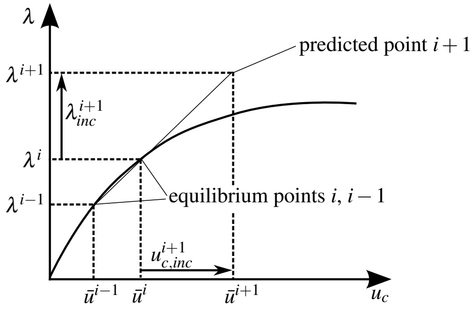


Figure 8.3: Predictor scheme for displacement controlled analysis

force determining the value of s is the ratio of the prescribed displacement increments in two subsequent load steps. Accordingly, a predictor used in context of optimization should be driven by the ration of design modifications for two subsequent optimization iterations. Design modifications can be measured by step length factors α used in the line search (see section 5.6), thus an appropriate predictor can be formulated as

$$\begin{aligned} \mathbf{u}^{i+1}_{inc} &= \frac{\alpha^{i+1}}{\alpha^{i}} \cdot \left(\mathbf{u}^{i} - \mathbf{u}^{i-1}\right) \\ \lambda^{i+1}_{inc} &= \frac{\alpha^{i+1}}{\alpha^{i}} \cdot \left(\lambda^{i} - \lambda^{i-1}\right) \end{aligned} \tag{8.3}$$

In case of a fixed step line search, it holds that $\frac{\alpha^{i+1}}{\alpha^{i}} = 1$. In this case, the predicted changes in displacement and load factor are identical to the changes observed in the last optimization step.

8.5 Considering of multiple load-path problems and avoiding of over-critical points

As already mentioned in section 8.1, bistable structures show a significant buckling behaviour. Buckling is a very complex mechanical process, especially when buckling of shell structures is considered, whereat the load path may be non-unique within the buckling occurrence. This phenomenon of non-unique load paths even can be observed for the simple two-bar-truss structure, when a steep truss structure is considered. Figure 8.4 shows such a steep two-bar truss with a height-to-span ratio of 2, where a single load is applied to the top node pointing in vertical displacement direction v. Structural analysis of this system leads to the following load displacement diagram (figure 8.5). The primary load path shows the typical snap-through characteristics with two limit points and an instable state of self equilibrium in between. Furthermore, a secondary load path exists, which separates suddenly from the primary path and shows a constant negative slope without the characteristic limit points. The point where the secondary load path separates from the primary load path is called bifurcation point. The structure's displacement $\mathbf{u}$ and the corresponding load factor are not uniquely related anymore for a given controlled displacement u_c beyond this point.

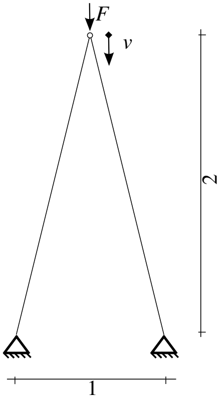


Figure 8.4: Statical system of steep two-bar truss

Figure 8.6 compares the deformations of the two-bar truss for both load paths. The primary load path shows the expected pure vertical deflection, ending up in the snap-through of the truss structure. The secondary load

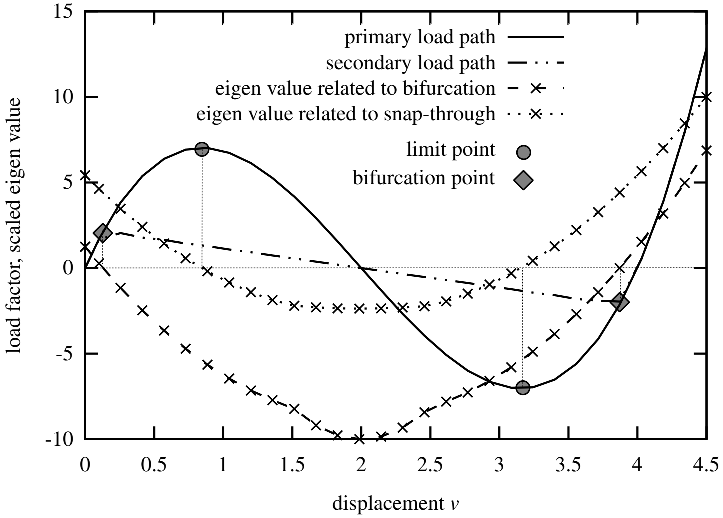


Figure 8.5: Load displacement curves and eigen values of two-bar truss problem

path, on the other hand, is related to horizontal buckling of the entire structure. This ambivalence is characteristic for stability problems. An "ideal" structure without any imperfection would stick to the primary load path in an instable state of equilibrium even for over critical loads, whereat a "real" structure including imperfections (for simulation purpose, imperfection was applied as a horizontal load component of $0.01 \cdot F$) would follow the secondary path instead.

Considering bistable structures, bifurcation and over critical load states are very important issues. Newly generated self stable states usually are not intended to be located in over critical regions, as ambivalent states of displacement at the specified self stable configuration would question the accessibility and robustness of this solution. For this purpose, it is important to be able to determine critical points and over critical regions within the load-displacement relation. Critical points, beneath limit and bifurca-

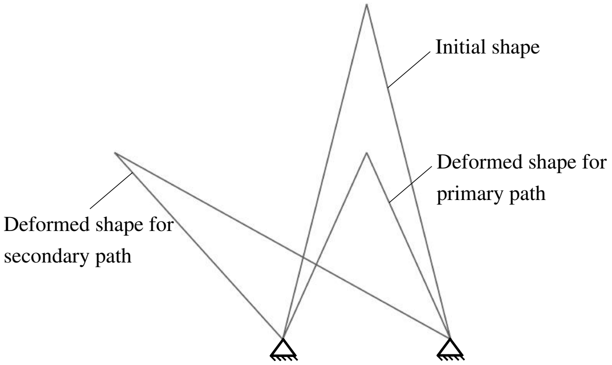

Initial shape
Deformed shape for primary path
Deformed shape for secondary path

Figure 8.6: Comparison of displacements on different load paths for two-bar truss problem

tion points, are characterized by a singular tangential stiffness matrix $\mathbf{K^T}$, so at least one eigen value of $\mathbf{K^T}$ has to be zero for a state of displacement describing a critical point [19]. Over critical states are characterized by the tangential stiffness matrix being indefinite [135]. The corresponding negative eigen values and related eigen vectors indicate an instable state of equilibrium, where the system would release stored strain energy if the deformation mode indicated by the eigen mode was stimulated. This behaviour is also reflected by figure 8.5, where both phenomena, the snap through as well as the bifurcation effect, are related to a specific eigen pair of the stiffness matrix. The numerically smaller eigen value is zero for both bifurcation points and negative for the instable states in between, whereat the second eigen value shows the same behaviour with respect to the snap through of the structure on the primary load path.

Coming back to bistable structures, it can be noted that a stable state of self equilibrium has to satisfy the condition of the tangential stiffness matrix being positive definite, where all eigen values have to be strictly positive. This condition ensures that the obtained self stable point is neither an instable configuration, nor it is located in a region of ambiguous load paths.

8.6 Response function and sensitivity analysis

As mentioned in sections 8.1 and 8.4, a bistable structure is characterized by multiple shapes the structure may assume without being subject to any loading. This property is utilized in order to formulate a response function which can be used for optimizing a structure with respect to bistable nature. The response function is based on a displacement controlled nonlinear static analysis. As the structure is meant to become self stable for specific values of controlled displacement, the absolute amount or the square of the load factor is minimized, such that the optimization problem can be written as

$$\begin{aligned} & min\ f(\mathbf{s},\bar{u}) = \|\lambda(\mathbf{s},\bar{u})\| \ \vee\ [\lambda(\mathbf{s},\bar{u})]^2 \\ & st \end{aligned} \tag{8.4a}$$

$$\mathbf{S}(\mathbf{s},\lambda,\mathbf{u}) = \mathbf{f_{int}}(\mathbf{u}) - \lambda \cdot \mathbf{f_{ext}} = \mathbf{0} \tag{8.4b}$$

$$C(\mathbf{s},\bar{u},\mathbf{u}) = u_c - \bar{u} = 0 \tag{8.4c}$$

where equation **S** is the residual formulation of the equilibrium condition and C describes the control equation for the displacement controlled path following algorithm, where u_c is the actual displacement at the controlled degree of freedom, and $\bar{u}$ is the assigned target value.

The sensitivity formulation of the objective function $\frac{df}{ds_i}$ is quite easy in this case, as the objective only depends on the design **s** and on the constant target displacement $\bar{u}$, but not on the state field **u**. Thus the total derivative is identical with the partial derivative with respect to s_i in this case:

$$\frac{df}{ds_i} = \frac{\partial f}{\partial s_i} + \frac{\partial f}{\partial \bar{u}} \cdot \underbrace{\frac{\partial \bar{u}}{\partial s_i}}_{0} \Rightarrow \frac{df}{ds_i} = \frac{\partial \|\lambda\|}{\partial s_i} \vee \frac{\partial \lambda^2}{\partial s_i} \tag{8.5}$$

In order to determine the partial derivative of the load factor, equations **S** and C have to be considered, too.
In a first step, equation **S** is derived with respect to s_i, in order to determine the derivative of the displacement field with respect to the design variable $\frac{\partial \mathbf{u}}{\partial s_i}$, similar to section 5.5.

$$\frac{d\mathbf{S}}{ds_i} = \underbrace{\frac{\partial \mathbf{f_{int}}}{\partial s_i} - \frac{\partial \left[\lambda \cdot \mathbf{f_{ext}}\right]}{\partial s_i}}_{\frac{\partial \mathbf{S}}{\partial s_i}} + \underbrace{\frac{\partial \mathbf{f_{int}}}{\partial \mathbf{u}}}_{\frac{\partial \mathbf{S}}{\partial \mathbf{u}}} \cdot \underbrace{\frac{\partial \mathbf{u}}{\partial s_i}}_{\frac{\partial \mathbf{u}}{\partial s_i}} = \mathbf{0}$$

$$\frac{\partial \mathbf{f_{int}}}{\partial s_i} - \lambda \cdot \frac{\partial \mathbf{f_{ext}}}{\partial s_i} - \frac{\partial \lambda}{\partial s_i} \cdot \mathbf{f_{ext}} + \mathbf{K^T} \cdot \frac{\partial \mathbf{u}}{\partial s_i} = \mathbf{0} \tag{8.6}$$

$$\frac{\partial \mathbf{u}}{\partial s_i} = \left[\mathbf{K^T}\right]^{-1} \cdot \left(\lambda \cdot \frac{\partial \mathbf{f_{ext}}}{\partial s_i} + \frac{\partial \lambda}{\partial s_i} \cdot \mathbf{f_{ext}} - \frac{\partial \mathbf{f_{int}}}{\partial s_i}\right)$$

A similar derivation can be found in Firl [52] for example. In this work a load controlled path following scheme was used, where the external load was considered to be constant, so that λ and all of its derivatives vanished accordingly.
In a second step, control equation C is considered. An operator vector $\mathbf{v}$ is defined at first, picking the controlled displacement u_c from the complete displacement vector $\mathbf{u}$ by scalar multiplication.

$$\begin{aligned} &C = u_c - \bar{u} = 0 \\ &C = \mathbf{v}^T \cdot \mathbf{u} - \bar{u} = 0 \\ &with \\ &\mathbf{v}^T = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \end{aligned} \tag{8.7}$$

As the controlled displacement u_c is set as a constant throughout the optimization, the derivative $\frac{dC}{ds_i}$ has to vanish.

$$
\begin{aligned}
&\frac{dC}{ds_i} = \frac{\partial C}{\partial s_i} + \frac{\partial C}{\partial \mathbf{u}} \cdot \frac{\partial \mathbf{u}}{\partial s_i} = 0 \\
&\underbrace{\left(\frac{\partial \mathbf{v}}{\partial s_i}\right)^T}_{\mathbf{0}} \cdot \mathbf{u} + \mathbf{v}^T \cdot \frac{\partial \mathbf{u}}{\partial s_i} + \mathbf{v}^T \cdot \frac{\partial \mathbf{u}}{\partial s_i} = 0 \\
&2 \cdot \mathbf{v}^T \cdot \frac{\partial \mathbf{u}}{\partial s_i} = 0 \\
&\mathbf{v}^T \cdot \frac{\partial \mathbf{u}}{\partial s_i} = 0
\end{aligned}
\tag{8.8}
$$

Usage of equation 8.6 delivers

$$
\frac{dC}{ds_i} = \underbrace{\mathbf{v}^T}_{\frac{\partial C}{\partial \mathbf{u}}} \cdot \underbrace{\left[\mathbf{K^T}\right]^{-1}}_{\left[\frac{\partial \mathbf{S}}{\partial \mathbf{u}}\right]^{-1}} \cdot \underbrace{\left(\lambda \cdot \frac{\partial \mathbf{f_{ext}}}{\partial s_i} + \frac{\partial \lambda}{\partial s_i} \cdot \mathbf{f_{ext}} - \frac{\partial \mathbf{f_{int}}}{\partial s_i}\right)}_{-\frac{\partial \mathbf{S}}{\partial s_i}} = 0 \tag{8.9}
$$

This equation now can be transfered to an adjoint formulation

$$
\left(\lambda \cdot \frac{\partial \mathbf{f_{ext}}}{\partial s_i} + \frac{\partial \lambda}{\partial s_i} \cdot \mathbf{f_{ext}} - \frac{\partial \mathbf{f_{int}}}{\partial s_i}\right)^T \cdot \underbrace{\left[\mathbf{K^T}\right]^{-1} \cdot \mathbf{v}}_{\Lambda} = 0 \tag{8.10}
$$

where Λ is the vector of adjoint variables which is constant for all design variables:

$$
\Lambda = \left[\mathbf{K^T}\right]^{-1} \cdot \mathbf{v} \tag{8.11}
$$

Using the adjoint variable vector, formula 8.10 can be solved with respect to $\frac{\partial \lambda}{\partial s_i}$:

$$
\frac{\partial \lambda}{\partial s_i} = \frac{\left(\frac{\partial \mathbf{f_{int}}}{\partial s_i}\right)^T \cdot \Lambda - \lambda \cdot \left(\frac{\partial \mathbf{f_{ext}}}{\partial s_i}\right)^T \cdot \Lambda}{\mathbf{f_{ext}}^T \cdot \Lambda} \tag{8.12}
$$

Having this information, the derivative of the actual response function according to equation 8.4a can be computed straight-forward.

8.7 Example 1: Hinged cylindrical roof

The first example considered in this context is the hinged cylindrical roof. This example is a very popular benchmark for nonlinear finite element shell formulations due to its snapping behaviour, and was published among others in Hughes, Liu [74] or Sze, Liu, Lo [124]. The uniaxially curved structure is a shallow cylinder segment which is hinge supported at the straight edges and free at the curved edges (see figure 8.7). The cylinder has a radius of $R = 2,540$ and a length of $L = 508$. The considered segment has an opening angle of $\phi = 0.2rad = 11.44°$. A single force is acting in the center point of the segment in radial direction, and two material setups of the structure will be investigated. For the first material setup, linear elastic isotropic material behaviour with an elastic modulus of $E = 3,102.75$ and a Poisson's Ratio of $\nu = 0.3$ is assumed. The shell thickness is set to $t = 12.6$ according to Sze, Liu, Lo [124]. In the second setup, a stacked sequence of orthotropic material will be considered with material properties according to figure 8.8.

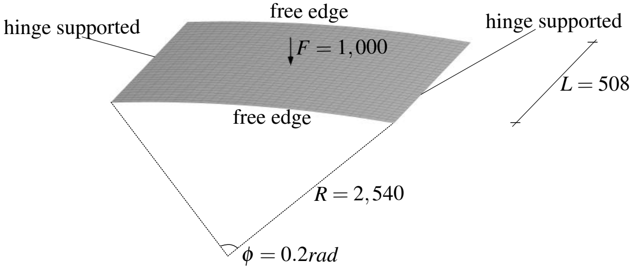


Figure 8.7: Statical system of hinged cylinder roof example

For the isotropic material setup, the structure can be expected to behave monostable [65]. In order to proof this assumption and to determine the behaviour of the orthotropic shell, static analysis of the initial structure is performed. The cylinder segment has got a rise of approximately 12.5, and analysis is performed until a maximum vertical deflection of

E_1	E_2	E_3	G_{12}	G_{13}	G_{23}	ν_{12}	ν_{13}	ν_{23}
$3,300$	$1,100$	$1,100$	660	550	550	0.25	0.25	0.25
		Stack sequence			0°/90°/0°			
		Ply thicknesses			4.2/4.2/4.2			
		Zero-degree-direction			Hoop direction			

Figure 8.8: Material specifications of orthotropic material setup

the loaded node of 30.0 is reached, in order to catch the complete snap through in the analysis. The related load-displacement curves are shown in figure 8.9, and it can be seen that the structure shows a highly non-linear

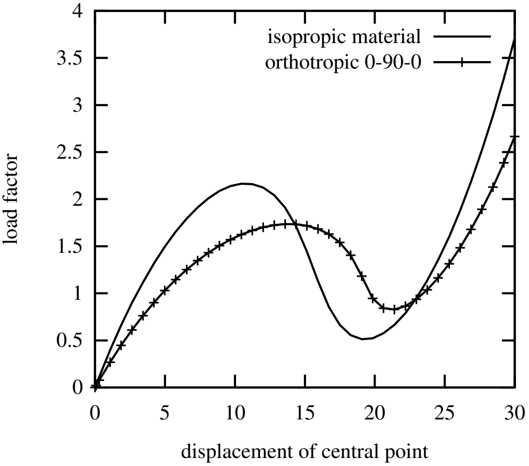


Figure 8.9: Load displacement curves of initial design of cylinder roof using different material setups

load carrying behaviour with a significant snap through characteristic, but the cylinder segment behaves mono-stable for both material setups, as the load factor is strictly positive for all deformed configurations.

8.7.1 Generation of a bistable configuration using pure shape optimization

In this first optimization, a bistable structure is to be generated using pure shape optimization. For this purpose, the response function introduced in section 8.6 is used. The newly generated state of self equilibrium should be characterized by a vertical displacement of the loaded node by 25. This is roughly the double rise of the shell segment, so that the structure is intended to behave self stable in the completely snapped through configuration, similar as the two-bar-truss introduced in section 8.1 does.
In the context of this optimization, all finite element nodes except the supported nodes and the neighboring nodes of the point of load application can be moved in surface normal direction. The supports and the area of load application are not taken into account as design space, in order to maintain the characteristics of the structure, such as the rise and the straight supported edges.

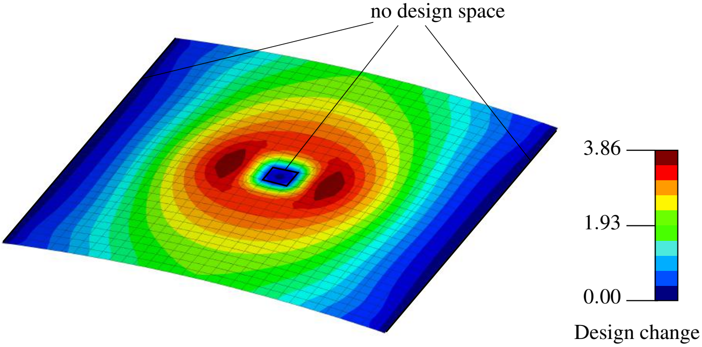


Figure 8.10: Design change of isotropic cylinder roof

The optimized design is shown in figures 8.10. It can be seen that the occurring design changes are comparable small, as the maximum design change is 3.86, which is just 30% of the shell thickness. This design modifications mainly happen in the center part of the roof segment, whereby local biaxial positive curvature is generated. Although the design changes are quite small, the influence onto the load carrying behaviour of the structure is tremendous, which is illustrated by the load displacement curve of the optimized roof segment (figure 8.11). The structure now behaves bistable, possessing a new self stable configuration for a displacement of the traced node of 24.932. At the same time, the maximum load bearing capacity of structure was increased from 2.16 to 3.3, as the central part of the roof now acts as a doubly curved structure, and so the stiffness in the pre-critical state is increased. As a consequence, the loss of stiffness in the deformation range from 12 to 20 is more severe, which can be seen in the steep drop of the load-displacement curve.

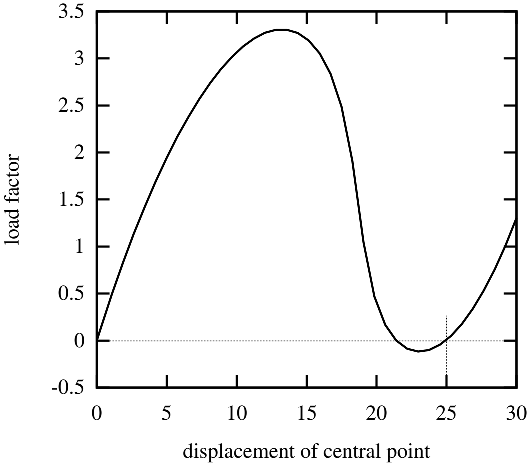


Figure 8.11: Load displacement curve of isotropic cylinder roof, optimized design

This example shows that the analysis and optimization procedure shown in sections 8.4 and 8.6 is able to generate new states of self equilibrium for a structure with a initial monostable snapping behaviour. As the behaviour of the considered system was quite good-natured in this example, no additional efforts with respect to ensuring regularity of the newly generated self stable configuration had to be made.

8.7.2 Generation of a bistable structure with predefined limit points

The optimization setup which was considered in section 8.7.1 is enhanced now. While in section 8.7.1 only a new self stable configuration was enforced, this example will also prescribe the snap through loads and limit point displacements which have to be overcome in order to change from one stable configuration to the other one. As this task makes a higher demand of the structure with several requirements to be met, a larger design space is used for optimization. For this reason, the orthotropic material setup of the cylinder roof is considered, and a combined shape and composite fiber orientation optimization is performed. Therefore three additional design variables are assigned to each finite element, controlling the material orientation in each ply. Further information about composite fiber optimization can be found in Masching et al [92] or Stegmann, Lund [122], for example. Again a self stable configuration for a traced displacement of 25 is requested, additionally the two limit points are specified for traced displacements of 10 and 22.5, with related limit load factors of 1.25 and -0.2. The limit point characteristics of these points are enforced by demanding the smallest eigen values of the tangential stiffness matrices to be equal to zero. In order to formulate and derive this eigen value constraint, a formulation based on Lindgaard, Lund [87] is used which approximates the displacement related derivatives of the tangential stiffness $\frac{\partial \mathbf{K}^{\mathbf{T}}}{\partial \mathbf{u}}$ matrix. These responses (load factor for $u_c = 25$, load factor for $u_c = 10$, load factor for $u_c = 22.5$, eigen value of tangential stiffness for $u_c = 10$ and eigen value of tangential stiffness for $u_c = 22.5$) are gathered in a least-squares formulation which finally is used as objective function within the optimization.

$$
\begin{aligned}
&min\ f = \\
&[\lambda\left(u_c = 25\right)]^2 + [\lambda\left(u_c = 10\right) - 1.25]^2 + [\lambda\left(u_c = 22.5\right) - (-0.2)]^2 + \\
&\left[eig\left(\mathbf{K}^{\mathbf{T}}\left(u_c = 10\right)\right)\right]^2 + \left[eig\left(\mathbf{K}^{\mathbf{T}}\left(u_c = 22.5\right)\right)\right]^2
\end{aligned}
\tag{8.13}
$$

The definition of shape optimization design variables is the same as the one which was used in section 8.7.1. The total number of design variables is 4086, 1014 nodal based design variables for shape optimization plus 3072 element based variables controlling fiber orientations. For the gradient filtering, a filter function with a rather small filter radius of 30 is used, leading to a filter function with four elements in radius.

The resultant geometry changes generated by the shape controlling design variables and the optimized fiber orientations for all three plies of the composite stack are shown in figures 8.12 and 8.13. The changes in geometry again are quite small, and in contrast to the isotropic optimization case, where the major changes with respect to geometry could be observed circularly, now the main shape modifications happen in generator direction

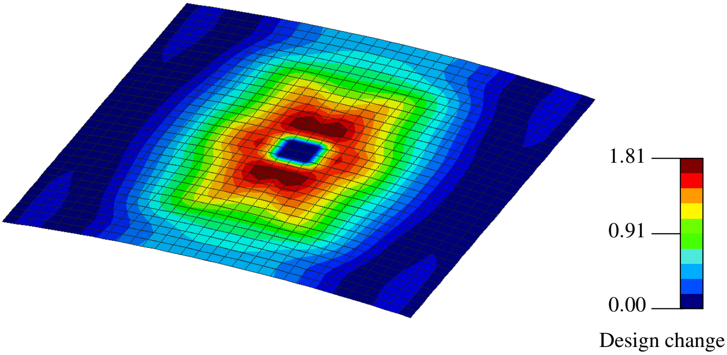


Figure 8.12: Design change of orthotropic cylinder roof

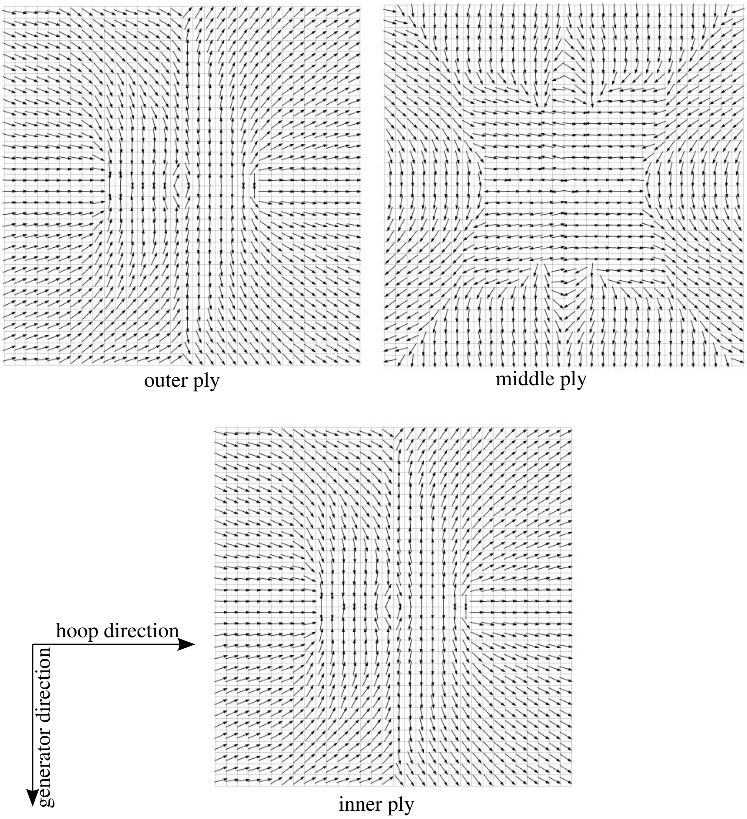


Figure 8.13: Optimized fiber orientation of orthotropic roof

around the central point of load application. Optimized fiber orientations (figure 8.13) show that three different regions of load carrying behaviour can be detected. Around the center of the supported lines, the initial fiber orientation, with the outer and inner ply being oriented in hoop direction and the middle ply oriented in un-curved generator direction, is maintained. In contrast, the preferred material orientation in the center area of the roof is inverted, so the inner and outer ply now are oriented in generator direction while fibers in the middle ply point in meridian direction. A third significant region can be observed close to the corner points of the roof, where the fibers of all three plies are equally oriented in diagonal direction.

Figure 8.14 shows the load-displacement curve of the optimized cylinder roof segment. The requested demands with respect to self-stability and limit points seem to be well fulfilled, and this impression is also confirmed by looking at the values of the sub-response functions being collected in the least-squares optimization (figure 8.15). The deviations from the requested load factors are ranging in magnitudes of 10^{-3}, and the zero eigen values of the stiffness matrices are met quite exactly, too.

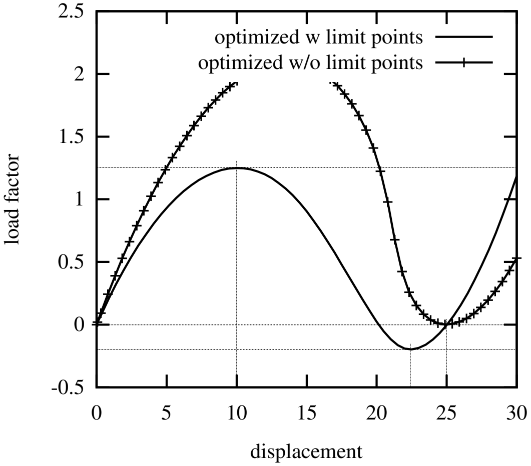


Figure 8.14: Load displacement curves of orthotropic cylinder roof, optimized designs with respect to different objectives

Figure 8.14 also presents the load displacement curve for an orthotropic roof segment being only optimized with respect to a self stable configura-

Structure response	Target value	Reached value
Load factor for $u_c = 25.0$	0.0	$2.66 \cdot 10^{-4}$
Load factor for $u_c = 10.0$	1.25	1.2508
Load factor for $u_c = 22.5$	-0.2	-0.1966
1st eigen value $\mathbf{K^T}$ for $u_c = 10.0$	0.0	$4.5 \cdot 10^{-6}$
1st eigen value $\mathbf{K^T}$ for $u_c = 22.5$	0.0	$6.2 \cdot 10^{-6}$

Figure 8.15: Least-squares optimization sub-responses of orthotropic cylinder roof

tion at $u_c = 25.0$ without further definitions of limit points, just as it was done for the isotropic roof in section 8.7.1. It is evident that this optimized design possesses the required self stable point at $u_c = 25.0$, but this point turns out to be a limit point at the same time. From the user's point of view, considering this structure to be a smart adaptive device in real life application, this property might be very unpleasant, as any occurring negative signed load would lead to an immediate snap-through of the structure, back towards the initial configuration. This awareness emphasizes that, for more complex structures, it may not be sufficient only to prescribe the self stable configurations of a structure. Defining additional limit points or constraining the allowable eigen values of the stiffness matrix in the self stable configuration may become essential in order to obtain a structure serving the intended purposes, as it was done in the first optimization using the response according to equation 8.13.

8.8 Example 2: Hyperbolic paraboloid

The previous example considered a singly curved cylindrical shell, and for this example the method of generating self stable states worked out very well. In this second example, a negatively doubly curved structure will be investigated. The system of consideration is a hyperbolic paraboloidal

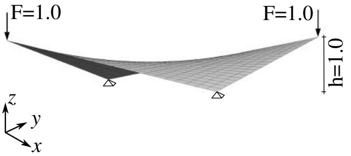


Figure 8.16: Isometric view of hyperbolic paraboloid system

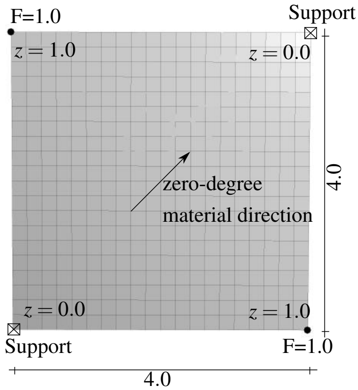


Figure 8.17: Top view of hyperbolic paraboloid system

shell which is loaded by two vertical forces at the high points and supported vertically at the low points (see figures 8.16 and 8.17). Additional support conditions are added such that the structure is supported statically determined. The shell forms a quite shallow hyperbolic paraboloidal structure with a footprint of $4.0 \cdot 4.0$ and a height of 1.0. The structure is modeled using composite material in a $90/0$ stagging sequence (bottom-to-top), where the zero-degree-direction is defined by the diagonal connecting the low points (see figure 8.17). Material properties are listed in figure 8.18, and the load-displacement curve of the shell is shown in figure 8.19, where the traced displacement is the vertical deflection of a loaded node.

The curve shows that the paraboloidical shell shows snap-through characteristics, but it does not behave bi- or multistable.

In order to optimize the hyperbolic paraboloid with respect to multistability, two additional self stable configurations are demanded, one for a traced displacement of $u_c = 1.0$ and the second one for a displacement of $u_c = 1.5$. Based on the information gained from the load-displacement curve in figure 8.19, the conclusion that the self stable configuration at

E_1	E_2	E_3	G_{12}	G_{13}	G_{23}
$2.0 \cdot 10^{10}$	$1.5 \cdot 10^{10}$	$1.5 \cdot 10^{10}$	$2.0 \cdot 10^{10}$	$1.0 \cdot 10^{10}$	$1.0 \cdot 10^{10}$

ν_{12}	ν_{13}	ν_{23}
0.20	0.20	0.20

Stack sequence	90°/0°
Ply thicknesses	0.05/0.05
Zero-degree-direction	Diagonal

Figure 8.18: Material specifications of orthotropic material setup

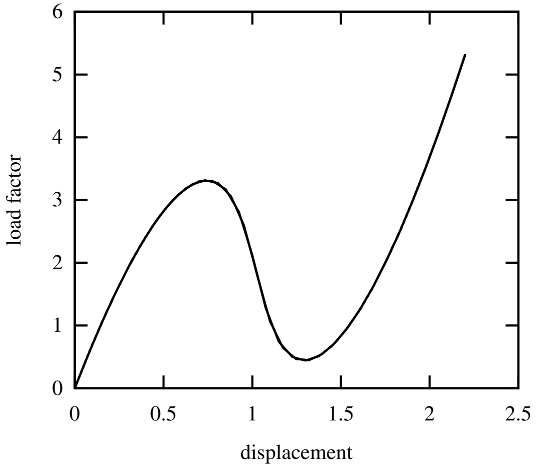


Figure 8.19: Initial load displacement diagram of hyperbolic paraboloidal shell, tracing the vertical deflection of a high point of the paraboloid

$u_c = 1.0$ will become an instable one is quite obvious. The purpose of prescribing this self stable configuration is to define the shape of the load-displacement-curve, although not all prescribed states might be reachable afterwards by a load controlled path following technique.

Using a corresponding least-squares objective function of the form

$$min\ f = [\lambda\,(u_c = 1.0)]^2 + [\lambda\,(u_c = 1.5)]^2 \tag{8.14}$$

the shell structure is optimized modifying its shape and its thickness. Thereby the high points as well as the low points of the paraboloid are not allowed to change their position, and thickness optimization is performed layer-wise per element. Resulting shape modifications and the final thickness distributions are shown in figures 8.20 and 8.21. Design

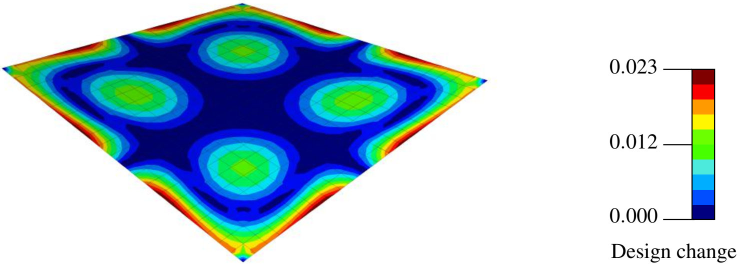


Figure 8.20: Shape changes of hyperbolic paraboloidal shell

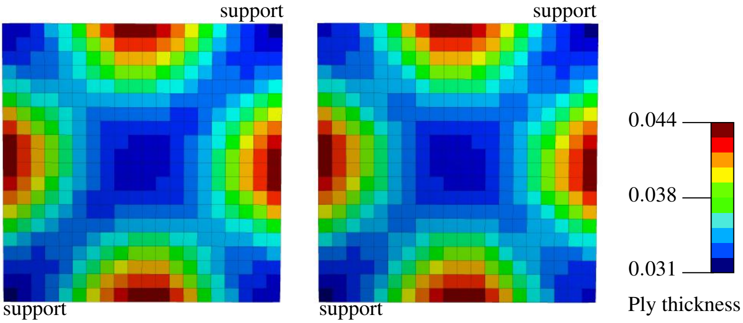


Figure 8.21: Hyperbolic paraboloid: Optimized ply thicknesses. Left: lower ply, Right: upper ply

changes are quite small ranging only up to one quarter of the initial shell thickness, and major changes occur in the middles of the free edges of the paraboloid. At the same time, the thickness of both plies is reduced over the entire structure, mainly along the diagonals of the structure. So the system's stiffness is reduced by thinning out the ply thicknesses along the main force trajectories in order to reduce the load factors related to the target displacements, while the load carrying behaviour is fine-tuned by the shape modifications along the edges leading to increased curvature.
Figure 8.22 shows the load displacement curve of the optimized structure. It can be seen that two additional self stable configurations have been created, whereat the targeted instable configuration at $u_c = 1.0$ is met almost perfect. The requested traced displacement of $u_c = 1.5$ for the second self stable configuration cannot be reached exactly, but the achieved value of $u_c = 1.4789$ is quite satisfying. Figure 8.22 also reveal that the optimized structure possesses multiple load paths. The primary path reflects the

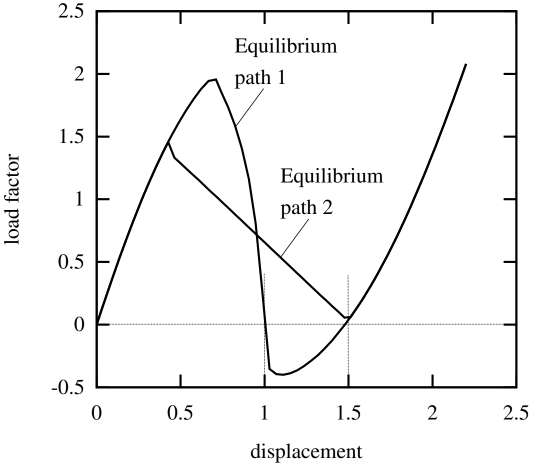


Figure 8.22: Load displacement diagram of optimized hyperbolic paraboloidal shell with multiple equilibrium paths

classical snap-through, while the secondary path bifurcates from the primary path below the limit load of the snap-through performing a skew and non-axial symmetric deformation due to an superimposed buckling mode. Figure 8.23 compares the deformed shapes of the structure structure for a traced displacement of 1.25. Unfortunately, the self stable configuration which was generated at $u_c = 1.4789$ is located in the over-critical regime of the load-displacement path, as the primary and secondary path reunite for a traced displacement $u_c > 1.5$. So the optimal solution violates the conditions which have been discussed in section 8.5.

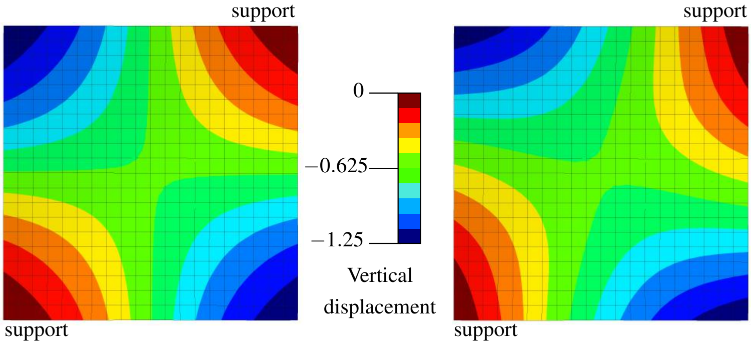


Figure 8.23: Vertical displacment of optimized paraboloid on primary (left) and secondary path (right)

In order to obtain an optimized design exhibiting a self stable equilibrium state for $u_c = 1.5$, which has to be a regular point at the same time, the optimization problem is modified. To this purpose, optimization starts from the design shown in figures 8.20 and 8.21 with a modified objective. within this optimization, only the deformation state $u_c = 1.5$ will be concerned, so the response function presented in equation 8.14 reduces to the second summand. Additionally, the eigen values of the tangential stiffness matrix at this state of deformation are constrained, in order to ensure the regularity of this state. The constraint is formulated by demanding the smallest eigen value to be larger than 1000. So the entire optimization problem formulation reads

$$\begin{aligned} & min\ f = [\lambda\,(\bar{u} = 1.5)]^2 \\ & st \\ & eig\left(\mathbf{K}^\mathbf{T}\,(u_c = 1.5)\right) > 1,000 \end{aligned} \tag{8.15}$$

The limit value for the smallest eigen value of $\mathbf{K}^\mathbf{T}$ is chosen based on specific model properties. As the eigen values of stiffness matrices depend on several input variable such as material parameters, shell thickness or dimensions of the investigated structure, it is difficult to determine uniform limit values in order to constrain stiffness matrix eigen values. In the particular example of consideration, the smallest eigen value of the elastic stiffness matrix is $2,500$, so a limit value of $1,000$ was used in order to constrain the eigen values of the tangential stiffness, being located in the same order of magnitude.

Figures 8.24 and 8.25 show the optimized design for the constrained problem, where the contours in figure 8.24 illustrate the changes compared to the un-optimized structure (figure 8.16). Comparing the results to the previous optimization, it can be seen that now the main changes with respect

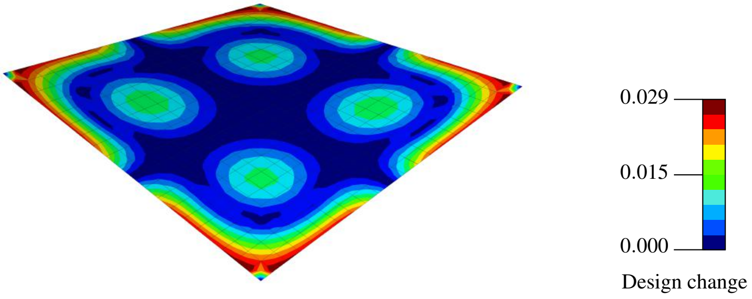


Figure 8.24: Shape changes of hyperbolic paraboloidal shell with eigenvalue constraint

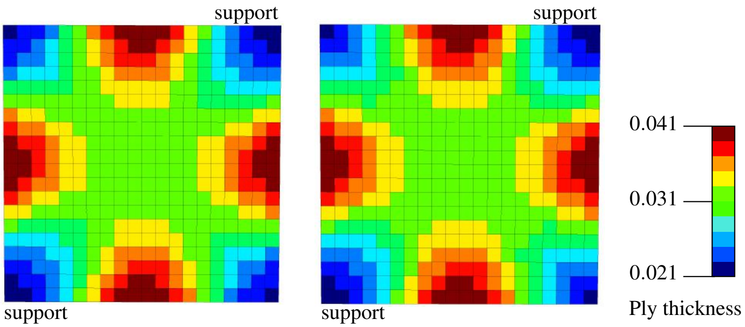


Figure 8.25: Hyperbolic paraboloid: Optimized ply thicknesses with eigenvalue constraint. Left: lower ply, Right: upper ply

to shape do not occur in the central regions of the edges, but close to the corners of the paraboloid. Comparison of optimized ply thicknesses reveals differences, too. While the unconstrained optimization showed a clear tendency to reduce material thicknesses along the diagonals of the structure(figure 8.21), the constrained optimization problem leads to constant ply thicknesses in the central region of the paraboloid, whereat thicknesses are locally decreased towards the corners and increased at the centers of the edges.

Figure 8.26 and 8.27 present the objective and constraint values as well as the load displacement curve for the final optimized structure. The ideal load factor of zero is almost reached, and the constraint with respect to the stiffness matrix eigen value is fulfilled. The load displacement curve

Load factor for $u_c = 1.5$	-0.035
1st eigen value $\mathbf{K^T}$ for $u_c = 1.5$	$1,003.6$

Figure 8.26: Least-squares optimization sub-responses of orthotropic cylinder roof

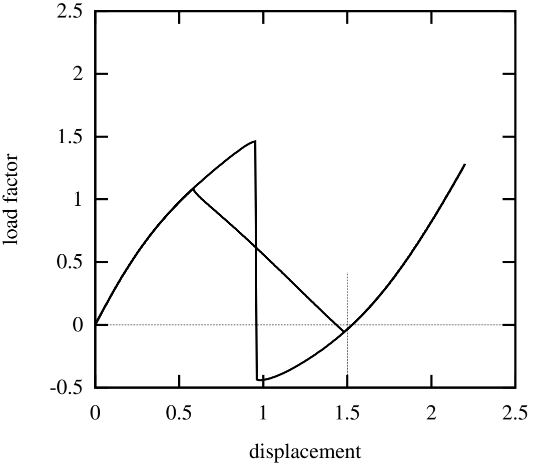


Figure 8.27: Load displacement diagram of optimized hyperbolic paraboloidal shell with eigenvalue constraint

reveals that the structure now possesses a self stable configuration very close to the target value of $u_c = 1.5$, which at the same time is a regular point now, showing a unique load-displacement-mapping.

8.9 Example 3: Three dimensional airfoil

The last example resumes the shape adaptive wing developed by Onur Bilgen [22] [23]. Here, the three dimensional airfoil with a variable chord length is considered, where the chord varies from $128mm$ to $177mm$ over a span on $248mm$ (figure 8.28). A four layer composite material setup according to figure 8.30 is assumed.
The wing is actuated into a shape providing high lift by two pressure loads acting along the entire span of the wing (figure 8.29), one at the trailing edge and one in the middle of the wing's bottom side. Due to this actuation, the wing assumes an S-bent configuration, similar to the 2D-model previously presented in section 7.3.3. The actuated configuration is dis-

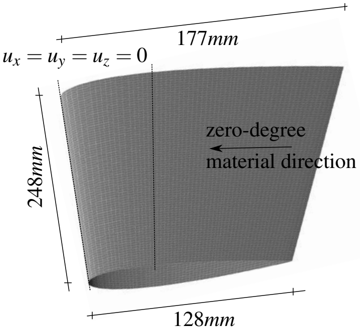


Figure 8.28: Geometry of adaptive wing

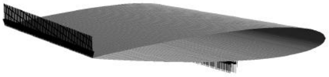


Figure 8.29: Actuation load of adaptive wing

E_1	E_2	E_3	G_{12}	G_{13}	G_{23}
290	180	180	145	90	90

ν_{12}	ν_{13}	ν_{23}
0.0	0.0	0.0

Stack sequence	0°/45°/−45°/90°
Ply thicknesses	0.5mm each

Figure 8.30: Material specifications of orthotropic material setup

played in figure 8.31, and the load displacement curve (figure 8.32) reveals that this actuation does not lead to a self stable configuration, as the load factor is strictly larger than zero for the entire actuation process.

For this wing structure, a combined ply thickness and fiber orientation optimization is performed. Goal of the optimization is to generate a structure which possesses a high lift configuration, similar to figure 8.31, as a state of self equilibrium. The underlying nonlinear static analysis is con-

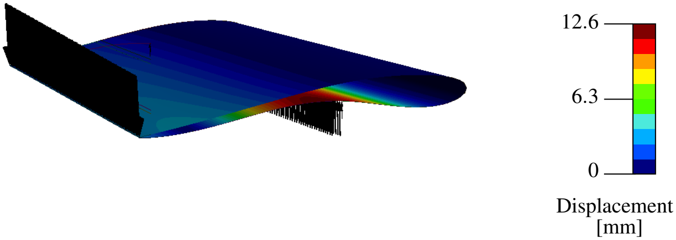


Figure 8.31: Initial wing design in actuated state, with actuation forces

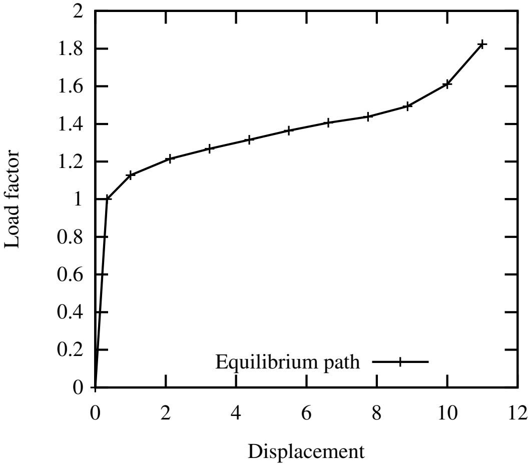


Figure 8.32: Load displacement curve of wing actuation. Traced displacement: Vertical deflection in center node of wing's bottom side

trolled via the vertical deflection of the center node of wing's bottom side $\bar{u}$. A value of $\bar{u} = 10$ is chosen in order to define the desired self stable configuration, and a load value of $\lambda = -0.075$ is requested in the negative load regime for a traced displacement of $\bar{u} = 7$, in order to ensure a sufficient stiffness of the structure and to avoid an unintended snap through of the structure. So the entire optimization problem can be formulated as a least squares problem:

$$min\ f = [\lambda\,(\bar{u} = 10)]^2 + [\lambda\,(\bar{u} = 7) + 0.075]^2 \qquad (8.16)$$

Eight design variables are defined for each finite element (four ply thickness variables, four fiber angle variables), so the optimization problem treats $80,000$ design variables in total. For each ply, a lower thickness bound of $t_{min} = 0.1mm$ has been set.

Optimization results are presented in figures 8.33 to 8.40. Thickness distributions (figures 8.33 to 8.36) show very similar results for all four plies.

Most thickness changes appear on the bottom side of the wing, which is not surprising as here the main part of the wing's deformation takes place. The ply thickness is reduced to t_{min} along the leading and the trailing edge as well as in the central area of the bottom side of the wing. These regions act as line hinges, facilitating the S-bending of the wing. Simultaneously, the ply thickness is increased at the long chord and on the top side of the

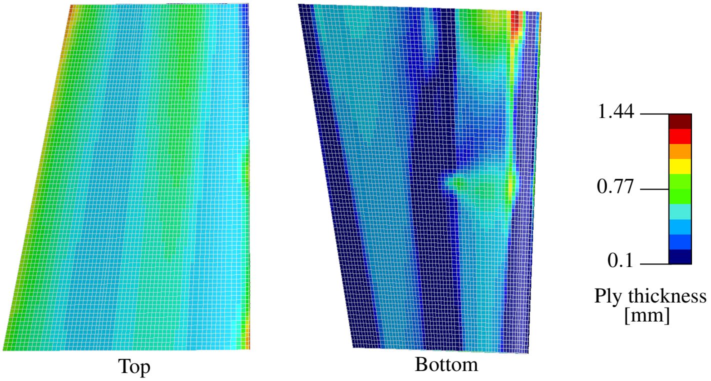


Figure 8.33: Thickness distribution for ply 1

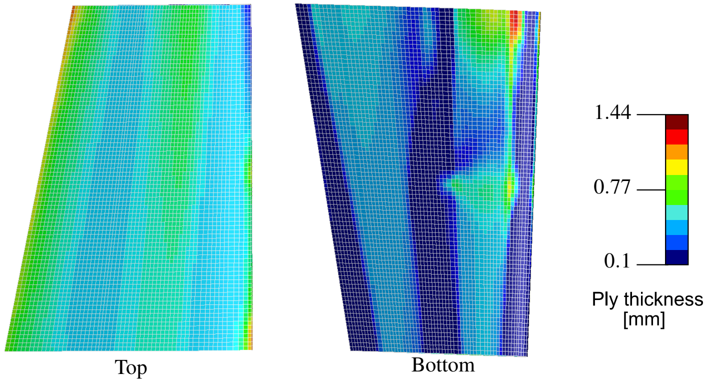


Figure 8.34: Thickness distribution for ply 2

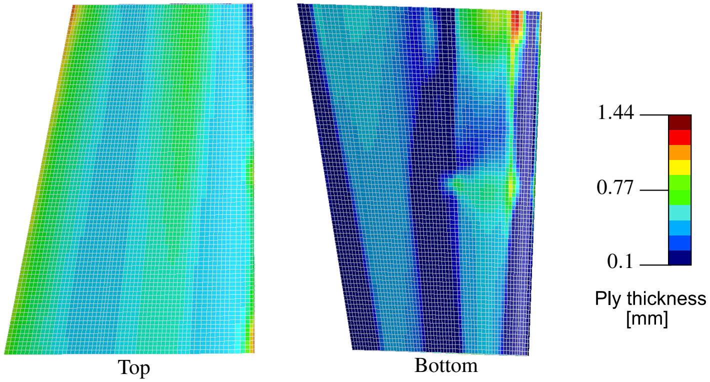


Figure 8.35: Thickness distribution for ply 3

wing along the trailing edge, in order to generate the desired stiffness of the structure.

Fiber orientations (figure 8.37 to 8.40) reflect the kinematic mechanisms already observed for the ply thickness distributions. Only few modifications happen on the wing's top side, while the fibers on the bottom side tend to be oriented in span direction in order to emphasis mechanism of

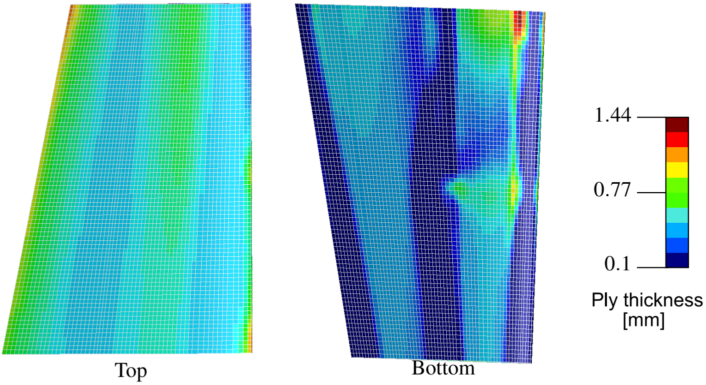


Figure 8.36: Thickness distribution for ply 4

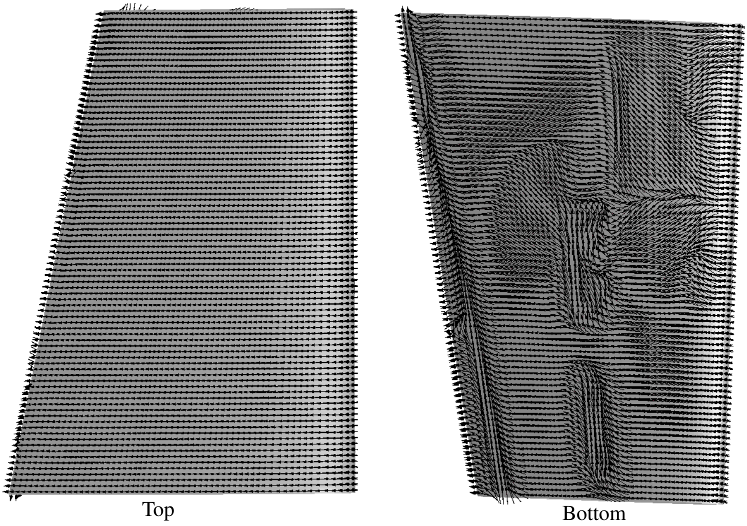
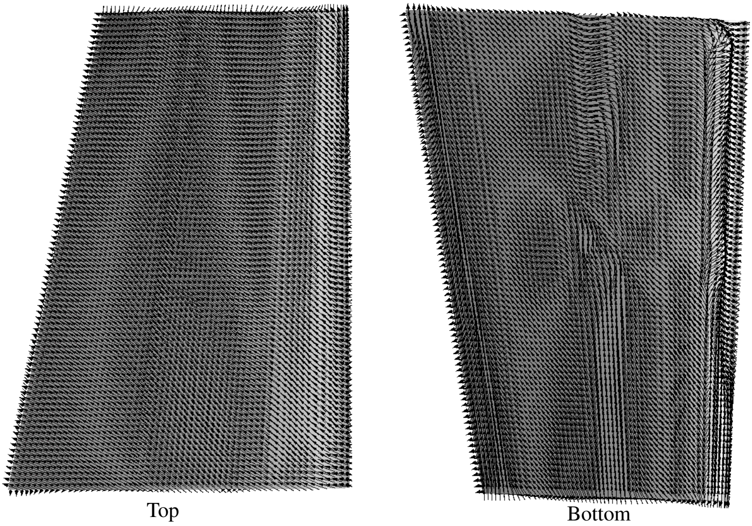


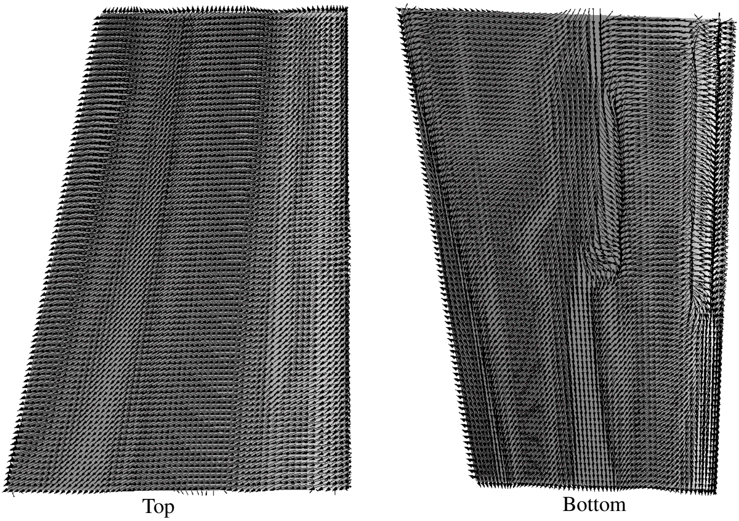
Figure 8.37: Fiber orientation for ply 1

the line hinges observed in the thickness distributions. Ply four shows an interesting material orientation on the top side of the wing. Here, the main part of the fibers towards the trailing edge is oriented in chord direc-


Top

Bottom

Figure 8.38: Fiber orientation for ply 2


Top

Bottom

Figure 8.39: Fiber orientation for ply 3

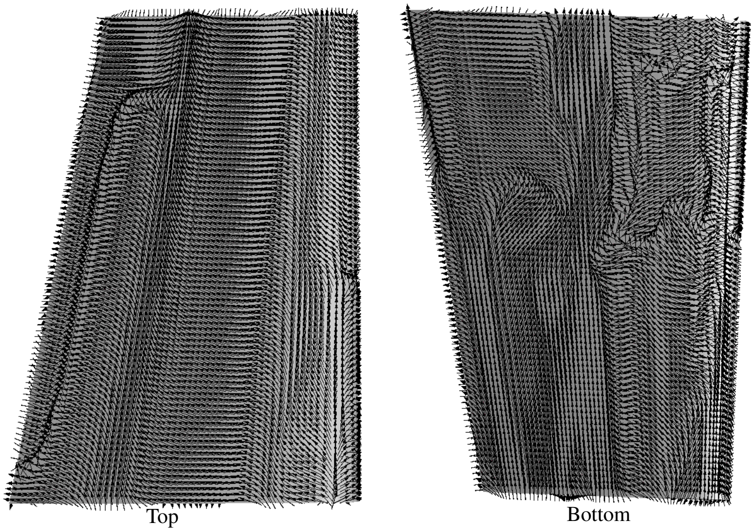


Figure 8.40: Fiber orientation for ply 4

tion, providing high stiffness. This stiff layout is interrupted by two fiber strings in span direction, again acting as line hinges.

The load displacement curve of the optimized airfoil (figure 8.41) shows that the optimization targets are met very well by the final design. The load factor $\lambda\ (\bar{u} = 10)$ is reduced to -0.001, and the load carrying behavior in the negative load regime is sufficiently high ($\lambda\ (\bar{u} = 7) = -0.076$). For a deformation of $\bar{u} = 10$, the load-displacement relation is well-defined, so the self-stable configuration can be assumed without the risk of ending up in an un-intended load path. The newly generated self-stable configuration is shown in figure 8.42, being very similar to the actuated shape of the initial wing design (figure 8.31).

In summary, the combined thickness and fiber orientation optimization succeeded in generating an airfoil possessing a self-stable high lift configuration.

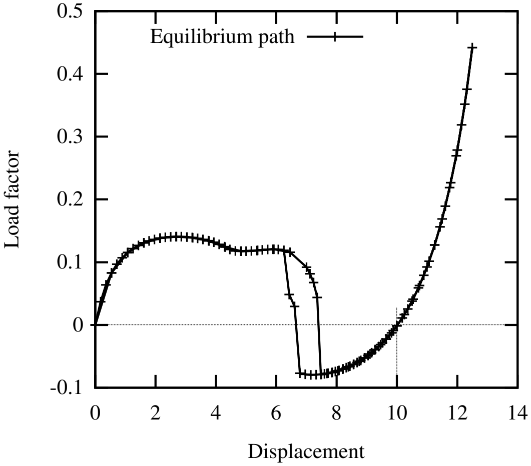


Figure 8.41: Load displacement curve of optimized wing design

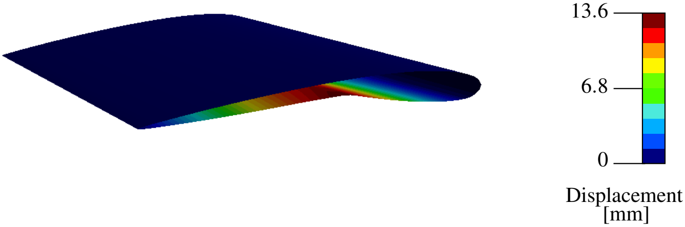


Figure 8.42: Optimized wing design in self-stable actuated state

8.10 Conclusion

In summary, it can be noted that the presented formulation of an optimization response function based on displacement controlled analysis is applicable for the generation and development of bistable shell structures. Avoidance of undesired or ambiguous solutions may require an enhanced formulation of the optimization problem, as it can be seen section 8.7.2 or in the last example.
The presented examples showed that small changes in shape may already have a high influence on the load carrying behaviour of a structure. Thus sensitivity of bistable structures with respect to imperfections has to be considered to be high, which also was observed by Coburn et al [43].

CHAPTER 9

Summary & Outlook

9.1 Summary

This thesis addressed the issue of generating structures with optimal kinematic properties, which is a central issue concerning controlled or selectively adapted structures. Thereby the main focus was on shell structures, where fundamental kinematic mechanisms of shells have been discussed and successfully generated by means of parameter free shape optimization. Depending on the loading, the stress state inside the structure and on restricting boundary conditions, line hinges or corrugated bellows turned out to be the design elements providing best flexibility. Results showed that the shapes of these kinematic elements may not be unique, especially when stress states dominated by normal forces are considered. Consequently the choice of different optimization setup parameters, such as shell thicknesses or filter radii, might lead to different optimal designs, whereby different designs may perform quite comparably with respect to the chosen objective function.

The gained awarenesses about kinematic mechanisms were transferred to actuated structures, and the issue of optimal actuation was discussed. This becomes an important aspect when optimal kinematic mechanisms are developed for smart structures, which also includes the problem of actuator placement and tuning. It turned out that optimization with respect to optimal actuation implies additional difficulties, as the parallel treatment of different types of design variables increases the non-convexity of the optimization problem tremendously. A modified filtering and shape control scheme was presented in order to improve orientation of the optimization algorithm in this highly non-convex design space.

Finally, the relatively new research field of bistable structures was addressed. Bistable structures might be an interesting alternative to "classical" smart structures, needing permanent actuator input. This thesis applied numerical optimization methods in order to generate structures exhibiting bistable character. For this purpose, a response function based on a displacement controlled nonlinear static analysis including an appropriate predictor scheme was developed. The fitness of this response function has been verified using single and double curved structures. Hereby one can observe that small changes in shape may lead to significant changes in the load carrying behaviour of shell structures. This especially appears when starting from uniaxially curved shells or when modifying the edge region of a shell.
As the investigated structures are characterized by a significant nonlinear load carrying behaviour, ensuring regularity of the target configuration is a crucial point in generating new self-stable configurations. For this purpose, an eigenvalue-based evaluation criteria was used in order to avoid solutions located on secondary load paths or within a buckling process.

9.2 Outlook

Starting from this thesis, continuative works should be performed in context of adaptive structures. Within this work, actuation was considered in an abstract way as a load onto the structure without regarding any constraints with respect to applicable actuator mechanisms, such as a limited adjustment travel or maximum allowable curvatures of a laminar actuator, for example. Enlarging the optimization process by such constraints would be a big step towards general applicability.

Another interesting point is verification of the shell designs optimized with respect to bistability. For this purpose, prototype construction and testing of selected shell designs could be performed, in order to verify computational results by experimental data.
Current research concerning bistable structures only addresses academic examples. Thus a proceeding step might be the application of bistability to a "real world"-example of a smart structure. As reliability and durability of this kind of structures is not tested or verified so far, one possible field of application might be unfolding mechanisms for satellites or other space structures requiring only a few or even only one transformation cycle in their entire life time.

CHAPTER A

Response functions and their derivatives

A.1 Geometric linear static computations

A.1.1 Derivation of the state equation

The state equation of static linear problems in structural mechanics can be formulated in the well-known form

$$\mathbf{S}(\mathbf{u}) = \mathbf{K} \cdot \mathbf{u} - \mathbf{f_{ext}} = \mathbf{0} \tag{A.1}$$

Taking the derivative of **S** with respect to a design variable s_i according to equation 5.31 requires the application of the chain rule of differentiation:

$$\begin{array}{ccccccccc} \frac{d\mathbf{S}}{ds_i} & = & \frac{\partial \mathbf{S}}{\partial s_i} & + & \frac{\partial \mathbf{S}}{\partial \mathbf{u}} & \cdot & \frac{\partial \mathbf{u}}{\partial s_i} & = & 0 \\ \frac{d\mathbf{S}}{ds_i} & = & \frac{\partial \mathbf{K}}{\partial s_i} \cdot \mathbf{u} - \frac{\partial \mathbf{f_{ext}}}{\partial s_i} & + & \mathbf{K} & \cdot & \frac{\partial \mathbf{u}}{\partial s_i} & = & 0 \end{array} \tag{A.2}$$

Solving this expression for $\frac{\partial \mathbf{u}}{\partial s_i}$ leads to

$$\begin{array}{ccccc} \frac{\partial \mathbf{u}}{\partial s_i} & = & \left[\frac{\partial \mathbf{S}}{\partial \mathbf{u}}\right]^{-1} & \cdot & \left[-\frac{\partial \mathbf{S}}{\partial s_i}\right] \\ \frac{\partial \mathbf{u}}{\partial s_i} & = & \mathbf{K^{-1}} & \cdot & \underbrace{\left[\frac{\partial \mathbf{f_{ext}}}{\partial s_i} - \frac{\partial \mathbf{K}}{\partial s_i} \cdot \mathbf{u}\right]}_{\widetilde{\mathbf{f}}} \end{array} \tag{A.3}$$

where $\widetilde{\mathbf{f}} = -\frac{\partial \mathbf{S}}{\partial s_i} = \frac{\partial \mathbf{f_{ext}}}{\partial s} - \frac{\partial \mathbf{K}}{\partial s} \cdot \mathbf{u}$ is the so-called *pseudo load vector*, which delivers the derivatives of the state variables $\mathbf{u}$ by being right-hand multiplied to $\mathbf{K}^{-1}$.

A.1.2 Strain energy

The strain energy of a linear problem can be formulated in the same way as the strain energy of a linear elastic spring:

$$E_{lin} = \frac{1}{2} \cdot \mathbf{u}^T \cdot \mathbf{f_{ext}} \tag{A.4}$$

Taking derivatives according to equation 5.29 and usage of equation A.3 leads to the direct sensitivity, showing all terms expected from the general formulation (equation 5.29):

$$\begin{aligned}
\frac{dE_{lin}}{ds_i} &= \frac{\partial E_{lin}}{\partial s_i} + \left[\frac{\partial E_{lin}}{\partial \mathbf{u}}\right]^T \cdot \frac{\partial \mathbf{u}}{\partial s_i} \\
\frac{dE_{lin}}{ds_i} &= \underbrace{\frac{1}{2} \cdot \mathbf{u}^T \cdot \frac{\partial \mathbf{f_{ext}}}{\partial s_i}}_{\frac{\partial E_{lin}}{\partial s_i}} + \underbrace{\left[\frac{1}{2} \cdot \mathbf{f_{ext}}\right]^T}_{\frac{\partial E_{lin}}{\partial \mathbf{u}}} \cdot \underbrace{\mathbf{K^{-1}}}_{\left[\frac{\partial \mathbf{S}}{\partial \mathbf{u}}\right]^{-1}} \cdot \underbrace{\left[\frac{\partial \mathbf{f_{ext}}}{\partial s_i} - \frac{\partial \mathbf{K}}{\partial s_i} \cdot \mathbf{u}\right]}_{-\frac{\partial \mathbf{S}}{\partial s_i}}
\end{aligned} \tag{A.5}$$

Following the thoughts about the symmetry of $\mathbf{K}$ and $\mathbf{K^{-1}}$ engaged in section 5.5, we can get to the adjoint formulation by permuting the vectors in the second summand:

$$\frac{dE_{lin}}{ds_i} = \underbrace{\frac{1}{2} \cdot \mathbf{u}^T \cdot \frac{\partial \mathbf{f_{ext}}}{\partial s_i}}_{\frac{\partial E_{lin}}{\partial s_i}} + \underbrace{\left[\frac{\partial \mathbf{f_{ext}}}{\partial s_i} - \frac{\partial \mathbf{K}}{\partial s_i} \cdot \mathbf{u}\right]^T}_{-\frac{\partial \mathbf{S}}{\partial s_i}} \cdot \underbrace{\mathbf{K^{-1}}}_{\left[\frac{\partial \mathbf{S}}{\partial \mathbf{u}}\right]^{-1}} \cdot \underbrace{\left[\frac{1}{2} \cdot \mathbf{f_{ext}}\right]}_{\frac{\partial E_{lin}}{\partial \mathbf{u}}} \tag{A.6}$$

Introducing the adjoint variables vector Λ, equation A.6 can be split in two parts, the first being independent of s_i and representing the computation of the adjoint variable, and the a second part considering the derivatives with respect to the design variable. For this response function, the adjoint variables vector turns out to be $\frac{1}{2} \cdot \mathbf{u}$.

$$\begin{aligned} \Lambda &= \underbrace{\mathbf{K}^{-1}}_{\left[\frac{\partial \mathbf{S}}{\partial \mathbf{u}}\right]^{-1}} \cdot \underbrace{\left[\frac{1}{2} \cdot \mathbf{f_{ext}}\right]}_{\frac{\partial E_{lin}}{\partial \mathbf{u}}} = \frac{1}{2} \cdot \mathbf{u} \\ \frac{dE_{lin}}{ds_i} &= \underbrace{\frac{1}{2} \cdot \mathbf{u}^T \cdot \frac{\partial \mathbf{f_{ext}}}{\partial s_i}}_{\frac{\partial E_{lin}}{\partial s_i}} + \underbrace{\left[\frac{\partial \mathbf{f_{ext}}}{\partial s_i} - \frac{\partial \mathbf{K}}{\partial s_i} \cdot \mathbf{u}\right]^T}_{-\frac{\partial \mathbf{S}}{\partial s_i}} \cdot \Lambda \end{aligned} \tag{A.7}$$

A.1.3 Nodal displacement

The consideration and optimization of a nodal displacement is a quite frequent task in optimization, especially when dealing with adaptive structures. In order to be able to formulate a response function D_{lin} focusing on a single displacement value, an operator vector $\mathbf{v}$ is introduced, which selects the desired degree of freedom from the state variable vector $\mathbf{u}$ by a scalar multiplication.

$$\begin{aligned} D_{lin} &= \mathbf{v}^T \cdot \mathbf{u} \\ \mathbf{v}^T &= [0 \ \cdots \ 0 \ 1 \ 0 \ \cdots \ 0] \end{aligned} \tag{A.8}$$

The vector $\mathbf{v}$ may also have more than one non-zero-entry if the desired displacement is evaluated in a direction which is not aligned parallel to a coordinate axis.

Computing the direct derivatives of D_{lin} is straight-forward repetition of the steps applied in section A.1.2, where the partial derivatives $\frac{\partial D_{lin}}{\partial s_i}$ vanish as $\mathbf{v}$ is a constant vector:

$$\begin{aligned}
\frac{dD_{lin}}{ds_i} &= \frac{\partial D_{lin}}{\partial s_i} + \left[\frac{\partial D_{lin}}{\partial \mathbf{u}}\right]^T \cdot \frac{\partial \mathbf{u}}{\partial s_i} \\
\frac{dD_{lin}}{ds_i} &= \underbrace{\mathbf{v^T}}_{\frac{\partial D_{lin}}{\partial \mathbf{u}}} \cdot \underbrace{\mathbf{K^{-1}}}_{\left[\frac{\partial \mathbf{S}}{\partial \mathbf{u}}\right]^{-1}} \cdot \underbrace{\left[\frac{\partial \mathbf{f_{ext}}}{\partial s_i} - \frac{\partial \mathbf{K}}{\partial s_i} \cdot \mathbf{u}\right]}_{-\frac{\partial \mathbf{S}}{\partial s_i}}
\end{aligned} \tag{A.9}$$

Again, the step to the adjoint formulation is a simple switch of operators. In contrast to the response function of strain energy, now the adjoint variable is an independent variable which cannot be computed by scaling the state variable.

$$\begin{aligned}
\Lambda &= \mathbf{K^{-1}} \cdot \mathbf{v} \\
\frac{dD_{lin}}{ds_i} &= \left[\frac{\partial \mathbf{f_{ext}}}{\partial s_i} - \frac{\partial \mathbf{K}}{\partial s_i} \cdot \mathbf{u}\right]^T \cdot \Lambda
\end{aligned} \tag{A.10}$$

A.1.4 A remark about responses based on linear analysis

The response functions presented so far were based on linear static analysis. As long as the state equation is linear, one also could avoid the application of the chain rule according to equation 5.29 by solving the state equation explicitly for the state variables $\mathbf{u} = \mathbf{K}^{-1} \cdot \mathbf{f_{ext}}$, and using this relation directly in the response function. By this mean, the derivatives with respect to **u** vanish and the chain rule becomes unneeded. This is shown exemplarily for the nodal displacement response:

$$\begin{aligned}
D_{lin} &= \mathbf{v}^T \cdot \mathbf{u} = \mathbf{v}^T \cdot \mathbf{K}^{-1} \cdot \mathbf{f_{ext}} \\
\frac{dD_{lin}}{ds_i} &= \mathbf{v}^T \cdot \left[\mathbf{K}^{-1} \cdot \frac{\partial \mathbf{f_{ext}}}{\partial s_i} + \frac{\partial \mathbf{K^{-1}}}{\partial s_i} \cdot \mathbf{f_{ext}}\right]
\end{aligned} \tag{A.11}$$

The derivative of an inverted matrix can be computed via the following relation [129]:

$$\begin{aligned}
&\mathbf{A}^{-1}\cdot\mathbf{A}=\mathbf{I}\\
&\mathbf{A}^{-1}\cdot\frac{\partial\mathbf{A}}{\partial s_i}+\frac{\partial\mathbf{A}^{-1}}{\partial s_i}\cdot\mathbf{A}=\mathbf{0}\\
&\frac{\partial\mathbf{A}^{-1}}{\partial s_i}=-\mathbf{A}^{-1}\cdot\frac{\partial\mathbf{A}}{\partial s_i}\cdot\mathbf{A}^{-1}
\end{aligned} \tag{A.12}$$

Applying equation A.12 to equation A.11 leads to

$$\begin{aligned}
\frac{dD_{lin}}{ds_i}&=\mathbf{v}^T\cdot\left[\mathbf{K}^{-1}\cdot\frac{\partial\mathbf{f_{ext}}}{\partial s_i}-\mathbf{K}^{-1}\cdot\frac{\partial\mathbf{K}}{\partial s_i}\cdot\underbrace{\mathbf{K}^{-1}\cdot\mathbf{f_{ext}}}_{\mathbf{u}}\right]\\
\frac{dD_{lin}}{ds_i}&=\mathbf{v}^T\cdot\mathbf{K}^{-1}\cdot\left[\frac{\partial\mathbf{f_{ext}}}{\partial s_i}-\frac{\partial\mathbf{K}}{\partial s_i}\cdot\mathbf{u}\right]
\end{aligned} \tag{A.13}$$

which is identical to equation A.9.

A.2 Geometric nonlinear static computations

A.2.1 Notation and state equation for nonlinear equilibrium conditions

In the previous section (A.1), linear problems were discussed, for which a linear dependency between the load $\mathbf{f_{ext}}$ and the displacement field $\mathbf{u}$ was existing. Due to this linear dependency, the consideration of different stages of loading was needless, as all load stages and the related displacement could be computed by scaling the properties of an arbitrary reference state. In the actual section, we consider a nonlinear residual equation of the form

$$\mathbf{S}(\mathbf{u},\lambda)=\mathbf{f_{int}}(\mathbf{u})-\lambda\cdot\mathbf{f_{ext}}=\mathbf{0} \tag{A.14}$$

which raises the necessity to distinguish different load stages of a reference external load $\mathbf{f_{ext}}$. For this purpose, the load factor λ is introduced, which scales the reference load vector.

In the following, it is assumed that λ and $\mathbf{u}$ form a pair which satisfies equation A.14.

A.2.2 Derivation of the state equation

In order to take the derivative of the state equation A.14 with respect to a design parameter s_i, again the chain rule of differentiation has to be applied.

$$\begin{array}{rcccccccl} \frac{d\mathbf{S}}{ds_i} & = & \frac{\partial \mathbf{S}}{\partial s_i} & + & \frac{\partial \mathbf{S}}{\partial \mathbf{u}} & \cdot & \frac{\partial \mathbf{u}}{\partial s_i} & = & 0 \\ \frac{d\mathbf{S}}{ds_i} & = & \frac{\partial \mathbf{f_{int}}}{\partial s_i} - \lambda \cdot \frac{\partial \mathbf{f_{ext}}}{\partial s_i} & + & \mathbf{K^T} & \cdot & \frac{\partial \mathbf{u}}{\partial s_i} & = & 0 \end{array} \tag{A.15}$$

exploiting that the derivative of the residual force $\frac{\partial \mathbf{S}}{\partial \mathbf{u}} = \frac{\mathbf{f_{int}}}{\partial \mathbf{u}}$ is equal to the tangential stiffness matrix $\mathbf{K^T}$.
Solving this equation for the derivative of state variables again leads to a linear problem with the partial derivatives of the residual equation with respect to the design parameter acting as a pseudo load $\widetilde{\mathbf{f}} = -\frac{\partial \mathbf{S}}{\partial s_i} = \lambda \cdot \frac{\partial \mathbf{f_{ext}}}{\partial s_i} - \frac{\partial \mathbf{f_{int}}}{\partial s_i}$:

$$\begin{array}{rcccc} \frac{\partial \mathbf{u}}{\partial s_i} & = & \left[\frac{\partial \mathbf{S}}{\partial \mathbf{u}}\right]^{-1} & \cdot & \left[-\frac{\partial \mathbf{S}}{\partial s_i}\right] \\ \frac{\partial \mathbf{u}}{\partial s_i} & = & \left[\mathbf{K^T}\right]^{-1} & \cdot & \underbrace{\left[\lambda \cdot \frac{\partial \mathbf{f_{ext}}}{\partial s_i} - \frac{\partial \mathbf{f_{int}}}{\partial s_i}\right]}_{\widetilde{\mathbf{f}}} \end{array} \tag{A.16}$$

A.2.3 Strain energy

When computing the strain energy E_{nonlin} of a system under consideration of nonlinear behavior, the load stagging and the nonlinear load-displacement relation have to be taken into account. In order to compute the strain energy stored in a system configuration which is defined by the load factor λ and the displacement field $\mathbf{u}$, energy has to be measured in an integral manner over the load staging parameter $\overline{\lambda}$

$$\widetilde{E} = \int\limits_{\overline{\lambda}=0}^{\lambda} \mathbf{f_{ext}}^T \cdot \mathbf{u}\left(\overline{\lambda}\right) d\overline{\lambda} \tag{A.17a}$$

$$E = \lambda \cdot \mathbf{f_{ext}}^T \cdot \mathbf{u} - \widetilde{E} \tag{A.17b}$$

where equation A.17a determines the complementary strain energy $\widetilde{E}$, and equation A.17b computes the effective strain energy E. In these formulas, λ and $\mathbf{u}$ are still assumed to be conjugated via equation A.14, as well as $\overline{\lambda}$ and $\mathbf{u}\left(\overline{\lambda}\right)$ are.

As the functional relation $\mathbf{u}\left(\overline{\lambda}\right)$ is not known analytically, the integral is evaluated using an approximation, where the linear relation

$$\mathbf{u}\left(\overline{\lambda}\right) \approx \frac{\overline{\lambda}}{\lambda} \cdot \mathbf{u} \tag{A.18}$$

is assumed. This kind of linear assumption is quite frequently used in context of geometrically nonlinear optimization and also can be found in [53] or [34]. This simplifies equations A.17a and A.17b to

$$E \approx \widetilde{E} \approx \frac{1}{2} \cdot \lambda \cdot \mathbf{f_{ext}}^T \cdot \mathbf{u} = E_{nonlin} \tag{A.19}$$

which is similar to equation A.4 in the linear case.

In figure A.1, the real strain energy measure according to equation A.17b and the approximated strain energy according to equation A.19 are compared using an exemplary load-displacement curve.

Using the approximated formulation according to equation A.19, the derivative can be computed in a similar way as it was done for the linear case (equation A.5)

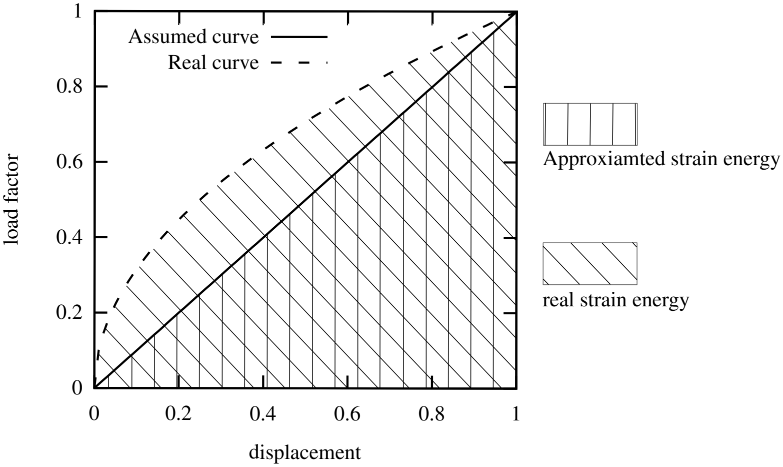


Figure A.1: Approximation of strain energy in geometrical nonlinear computations

$$\begin{aligned} \frac{dE_{nonlin}}{ds_i} &= \frac{\partial E_{nonlin}}{\partial s_i} + \left[\frac{\partial E_{nonlin}}{\partial \mathbf{u}}\right]^T \cdot \frac{\partial \mathbf{u}}{\partial s_i} \\ \frac{dE_{nonlin}}{ds_i} &= \underbrace{\frac{1}{2}\cdot \mathbf{u}^T\cdot\lambda\cdot\frac{\partial \mathbf{f_{ext}}}{\partial s_i}}_{\frac{\partial E_{nonlin}}{\partial s_i}} + \underbrace{\left[\frac{1}{2}\cdot\lambda\cdot\mathbf{f_{ext}}\right]^T}_{\frac{\partial E_{nonlin}}{\partial \mathbf{u}}} \cdot \underbrace{\left[\mathbf{K^T}\right]^{-1}}_{\left[\frac{\partial \mathbf{S}}{\partial \mathbf{u}}\right]^{-1}} \cdot \underbrace{\left[\lambda\cdot\frac{\partial \mathbf{f_{ext}}}{\partial s_i} - \frac{\partial \mathbf{f_{int}}}{\partial s_i}\right]}_{-\frac{\partial \mathbf{S}}{\partial s_i}} \end{aligned} \tag{A.20}$$

Switching to an adjoint formulation, we again end up with a two-step formulation, the first one giving the adjoint variables and the second one the derivative itself.

$$\begin{aligned}
\Lambda &= \underbrace{\left[\mathbf{K^T}\right]^{-1}}_{\left[\frac{\partial \mathbf{S}}{\partial \mathbf{u}}\right]^{-1}} \cdot \underbrace{\left[\frac{1}{2}\cdot\lambda\cdot\mathbf{f_{ext}}\right]}_{\frac{\partial E_{nonlin}}{\partial \mathbf{u}}} \\
\frac{dE_{nonlin}}{ds_i} &= \underbrace{\frac{1}{2}\cdot\mathbf{u}^T\cdot\lambda\cdot\frac{\partial\mathbf{f_{ext}}}{\partial s_i}}_{\frac{\partial E_{nonlin}}{\partial s_i}} + \underbrace{\left[\lambda\cdot\frac{\partial\mathbf{f_{ext}}}{\partial s_i} - \frac{\partial\mathbf{f_{int}}}{\partial s_i}\right]^T}_{-\frac{\partial \mathbf{S}}{\partial s_i}} \cdot\Lambda
\end{aligned} \tag{A.21}$$

A.2.4 Nodal displacement

The objective function focusing on a nodal displacement or rotation is formulated in the same way as it is done in the geometrical linear case. Again the operator vector **v** is multiplied to the displacement field vector **u** in order to define the objective function:

$$\begin{aligned}
D_{nonlin} &= \mathbf{v}^T\cdot\mathbf{u} \\
\mathbf{v}^T &= [0 \;\cdots\; 0 \;1\; 0 \;\cdots\; 0]
\end{aligned} \tag{A.22}$$

Computing the derivative, the nonlinear formulation behaves differently to the linear one, as the derivative of the state variables $\frac{\partial \mathbf{u}}{\partial s_i}$ is based on a different state equation (equation A.16).

$$\begin{aligned}
\frac{dD_{nonlin}}{ds_i} &= \frac{\partial D_{nonlin}}{\partial s_i} + \left[\frac{\partial D_{nonlin}}{\partial \mathbf{u}}\right]^T \cdot \frac{\partial \mathbf{u}}{\partial s_i} \\
\frac{dD_{nonlin}}{ds_i} &= \underbrace{\mathbf{v^T}}_{\frac{\partial D_{nonlin}}{\partial \mathbf{u}}} \cdot \underbrace{\left[\mathbf{K^T}\right]^{-1}}_{\left[\frac{\partial \mathbf{S}}{\partial \mathbf{u}}\right]^{-1}} \cdot \underbrace{\left[\lambda\cdot\frac{\partial\mathbf{f_{ext}}}{\partial s_i} - \frac{\partial\mathbf{f_{int}}}{\partial s_i}\right]}_{-\frac{\partial \mathbf{S}}{\partial s_i}}
\end{aligned} \tag{A.23}$$

Switching to the adjoint formulation, another similarity to the geometrical linear formulated response appears, as in both cases the right hand side in the adjoint equation system is formed by the operator vector.

$$\begin{aligned} \Lambda &= \left[\mathbf{K^T}\right]^{-1} \cdot \mathbf{v} \\ \frac{dD_{nonlin}}{ds_i} &= \left[\lambda \cdot \frac{\partial \mathbf{f_{ext}}}{\partial s_i} - \frac{\partial \mathbf{f_{int}}}{\partial s_i}\right]^T \cdot \Lambda \end{aligned} \tag{A.24}$$

Bibliography

[1] J. Adamy. *Nichtlineare Regelungen*. Berlin and Heidelberg: Springer, 2009. ISBN: 978-3642007934.

[2] G. Akhras. "Smart materials and smart systems for the future". In: *Canadian Military Journal 2000*.

[3] U. Andelfinger and E. Ramm. "EAS-elements for two-dimensional, three-dimensional, plate and shell structures and their equivalence to HR-elements". In: *International Journal for Numerical Methods in Engineering* 36.8 (1993), pp. 1311–1337. ISSN: 0029-5981. DOI: 10.1002/nme.1620360805.

[4] T. Arens et al. *Mathematik*. 1. Aufl. Heidelberg: Spektrum Akademischer Verlag, 2008. ISBN: 3827417589.

[5] J. H. Argyris. *Die Methode der finiten Elemente in der elementaren Strukturmechanik*. Braunschweig: Vieweg, 1986. ISBN: 3528089199.

[6] J. H. Argyris and H.-P. Mlejnek. *Computerdynamik der Tragwerke*. [Studienausg.] Vol. Studienbuch für Ingenieure und Naturwissenschaftler / John Argyris; Hans-Peter Mlejnek ; Bd. 3. Die Methode der finiten Elemente. Braunschweig [u.a.]: Vieweg, 1997. ISBN: 3528069163.

[7] L. Armijo. "Minimization of functions having Lipschitz continuous first partial derivatives". In: *Pacific J. Math* 16 (1966).

[8] J. S. Arora. *Introduction to optimum design*. 3rd ed. Boston and MA: Academic Press, 2011. ISBN: 978-0123813756.

[9] D. Ashlock. *Evolutionary computation for modeling and optimization*. New York: Springer, 2006. ISBN: 0387221964.

[10] H. Baier, C. Seesselberg, and B. Specht. *Optimierung in der Strukturmechanik*. Braunschweig and Wiesbaden: Vieweg, 1994. ISBN: 3528088990.

[11] R. Baldick. *Applied optimization: Formulation and algorithms for engineering systems*. Cambridge, U.K, and New York: Cambridge University Press, 2006. ISBN: 0521100283.

[12] B. Barthelemy, C. T. Chon, and R. T. Haftka. “Accuracy problems associted with semi-analytical derivatives of static response”. In: *Finite Elements in Analysis and Design* 4.3 (1988), pp. 249–265. ISSN: 0168874X. DOI: 10.1016/0168-874X(88)90011-X.

[13] F.-J. Barthold and E. Stein. “A continuum mechanical-based formulation of the variational sensitivity analysis in structural optimization. Part I: analysis”. In: *Structural Optimization* 11.1-2 (1996), pp. 29–42. ISSN: 0934-4373. DOI: 10.1007/BF01279652.

[14] K.-J. Bathe. *Finite-Elemente-Methoden*. 2., vollst. neu bearb. und erw. Aufl. Berlin [u.a.]: Springer, 2002. ISBN: 978-3540668060.

[15] K.-J. Bathe and E. Dvorkin. “A formulation of general shell elements—the use of mixed interpolation of tensorial components”. In: *Int J Numer Methods Eng* 22 (1986), pp. 697–722.

[16] K.-J. Bathe and E. Dvorkin. “A four-node plate bending element based on Mindlin/Reissner plate theory and a mixed interpolation”. In: *Int J Numer Methods Eng* 21 (1985), pp. 367–383.

[17] J.-L. Batoz and G. Dhatt. “Incremental displacement algorithms for nonlinear problems”. In: *International Journal for Numerical Methods in Engineering* 14.8 (1979), pp. 1262–1267. ISSN: 0029-5981. DOI: 10.1002/nme.1620140811.

[18] R. Bellman. *Introduction to matrix analysis*. 2nd ed. Vol. 19. Classics in applied mathematics. Philadelphia and Pa: Society for Industrial and Applied Mathematics (SIAM 3600 Market Street Floor 6 Philadelphia PA 19104), 1997. ISBN: 0898713994. URL: http://epubs.siam.org/ebooks/siam/classics_in_applied_mathematics/cl19.

[19] T. Belytschko, W. K. Liu, and B. Moran. *Nonlinear finite elements for continua and structures*. Chichester and New York: Wiley, 2000. ISBN: 0471987743.

[20] M. P. Bendsoe and O. Sigmund. *Topology optimization: Theory, methods, and applications*. Berlin and New York: Springer, 2003. ISBN: 3540429921.

[21] J.-M. Berthelot. *Composite Materials: Mechanical Behavior and Structural Analysis*. Mechanical Engineering Series. New York and NY: Springer New York, 1999. ISBN: 978-1-4612-0527-2.

[22] O. Bilgen, E. Flores, and M. I. Friswell. “Optimization of Surface-Actuated Piezocomposite Variable-Camber Morphing Wings”. In: *ASME 2011 Conference on Smart Materials, Adaptive Structures and Intelligent Systems* (2011), pp. 315–322.

[23] O. Bilgen and M. I. Friswell. “Piezoceramic composite actuators for a solid-state variable-camber wing”. In: *Journal of Intelligent Material Systems and Structures* 25 (2014), pp. 806–817.

[24] M. Bischoff. *Theorie und Numerik einer dreidimensionalen Schalenformulierung*. Bericht Nr. 30, Institut für Baustatik, Universität Stuttgart, 1999.

[25] K.-U. Bletzinger. “A consistent frame for sensitivity filtering and the vertex assigned morphing of optimal shape”. In: *Structural and Multidisciplinary Optimization* (2014). ISSN: 1615-147X. DOI: 10.1007/s00158-013-1031-5.

[26] K.-U. Bletzinger, M. Firl, and F. Daoud. “Approximation of derivatives in semi-analytical structural optimization”. In: *Computers & Structures* 86.13-14 (2008), pp. 1404–1416. ISSN: 00457949. DOI: 10.1016/j.compstruc.2007.04.014.

[27] K.-U. Bletzinger, M. Firl, and M. Fischer. “Parameter free shape design of thin shells: Efficient and effective, parallel solution techniques for very large design problems”. In: *2nd International Conference on Engineering Optimization, Lisbon* (2010).

[28] K.-U. Bletzinger et al. “Computational methods for form finding and optimization of shells and membranes”. In: *Computer Methods in Applied Mechanics and Engineering* 194.30-33 (2005), pp. 3438–3452. ISSN: 00457825. DOI: 10.1016/j.cma.2004.12.026.

[29] K.-U. Bletzinger et al. “Optimal shapes of mechanically motivated surfaces”. In: *Computer Methods in Applied Mechanics and Engineering* 199.5-8 (2010), pp. 324–333. ISSN: 00457825. DOI: 10.1016/j.cma.2008.09.009.

[30] K.-U. Bletzinger et al. “Shape Optimization with Program CARAT”. In: *Software systems for structural optimization*. Ed. by Hörnlein, H. R. E. M and K. Schittkowski. Vol. vol. 110. ISNM. Basel and Boston: Birkhäuser, 1993, pp. 97–124. ISBN: 978-3-7643-2836-8.

[31] P. T. Boggs and J. W. Tolle. “Sequential quadratic programming”. In: *Acta numerica* 4 (1995), pp. 1–51.

[32] M. Braun. *Nichtlineare Analyse von geschichteten, elastischen Flächentragwerken*. Bericht Nr. 19, Institut für Baustatik, Universität Stuttgart, 1995.

[33] C. G. Broyden. "The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations". In: *IMA Journal of Applied Mathematics* 6.1 (1970), pp. 76–90. ISSN: 0272-4960.

[34] T. Buhl, C. B. Pedersen, and O. Sigmund. "Stiffness design of geometrically nonlinear structures using topology optimization". In: *Structural and Multidisciplinary Optimization* 19 (2000), pp. 93–104.

[35] C. R. Calladine. *Theory of shell structures*. Cambridge [Cambridgeshire] and New York: Cambridge University Press, 1983. ISBN: 0521369452.

[36] Y. Cao, S. Li, and L. Petzold. "Adjoint sensitivity analysis for differential-algebraic equations: algorithms and software". In: *Journal of Computational and Applied Mathematics* 149 (2002), pp. 171–191.

[37] E. Carmel and D. Cohen-Or. "Warp-guided object-space morphing". In: *The Visual Computer* 13.9-10 (1998), pp. 465–478. ISSN: 01782789. DOI: `10.1007/s003710050118`.

[38] Z. Chen et al. "Nonlinear Geometric Effects in Mechanical Bistable Morphing Structures". In: *Physical Review Letters* 109 (2012), p. 114302.

[39] K. Choi and N. H. Kim. *Structural sensitivity analysis and optimization 1: Linear Systems*. Mechanical Engineering Series. New York: Springer Science+Business Media, 2005. ISBN: 0-387-23336-9.

[40] K. Choi and N. H. Kim. *Structural sensitivity analysis and optimization 2: Nonlinear Systems and Applications*. Mechanical Engineering Series. New York: Springer Science+Business Media, 2005. ISBN: 978-0-387-27169-9.

[41] K. K. Choong and E. Ramm. “Simulation of buckling process of shells by using the finite element method”. In: *Thin-Walled Structures* 31.1-3 (1998), pp. 39–72. ISSN: 02638231. DOI: `10.1016/S0263-8231(98)00002-0`.

[42] R. L. Clark, W. R. Saunders, and G. P. Gibbs. *Adaptive structures: Dynamics and control*. New York: Wiley, 1998. ISBN: 0471122629.

[43] B. H. Coburn et al. “Tristability of an orthotropic doubly curved shell”. In: *Composite Structures* 96 (2013), pp. 446–454. ISSN: 02638223. DOI: `10.1016/j.compstruct.2012.08.026`.

[44] B. Cockburn, G. Karniadakis, and C.-W. Shu. *Discontinuous Galerkin methods: Theory, computation, and applications*. Vol. 11. Lecture notes in computational science and engineering. Berlin and New York: Springer, 2000. ISBN: 3540667873.

[45] W. Dahmen and A. Reusken. *Numerik für Ingenieure und Naturwissenschaftler*. 2., korrigierte Aufl. Springer-Lehrbuch. Berlin and Heidelberg: Springer, 2008. ISBN: 978-3-540-76492-2.

[46] F. Daoud. *Formoptimierung von Freiformschalen: Mathematische Algorithmen und Filtertechniken*. Dissertation at the Chair of Structural Analysis at the Technical University of Munich, 2005.

[47] M. C. Delfour and J. P. Zolésio. *Shapes and geometries: Analysis, differential calculus, and optimization*. Advances in design and control. Philadelphia: Society for Industrial and Applied Mathematics, 2001. ISBN: 0898714893.

[48] P. Deuflhard. *Newton methods for nonlinear problems: Affine invariance and adaptive algorithms*. Vol. 35. Springer series in computational mathematics. Berlin and New York: Springer, 2004. ISBN: 3540210997.

[49] D. A. Di Pietro and A. Ern. *Mathematical aspects of discontinuous galerkin methods*. Vol. 69. Mathématiques et Applications. Berlin and New York: Springer, 2012. ISBN: 978-3-642-22979-4.

[50] M. Drela. "XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils". In: *Low Reynolds Number Aerodynamics*. Ed. by C. A. Brebbia et al. Vol. 54. Lecture Notes in Engineering. Berlin and Heidelberg: Springer Berlin Heidelberg, 1989, pp. 1–12. ISBN: 978-3-540-51884-6. DOI: 10.1007/978-3-642-84010-4_1.

[51] K. A. Dunn et al. "Application of Smart Materials/Technology at the Savannah River Site". In: *1999 Symposium on Smart Structures and Materials*. SPIE Proceedings. SPIE, 1999, pp. 64–75. DOI: 10.1117/12.348690.

[52] M. Firl. *Optimal Shape Design of Shell Structures*. Dissertation at the Chair of Structural Analysis at the Technical University of Munich, 2010.

[53] M. Firl and K.-U. Bletzinger. "Shape optimization of thin walled structures governed by geometrically nonlinear mechanics". In: *Computer Methods in Applied Mechanics and Engineering* 237-240 (2012), pp. 107–117. ISSN: 00457825. DOI: 10.1016/j.cma.2012.05.016.

[54] M. Firl, R. Wüchner, and K.-U. Bletzinger. "Regularization of shape optimization problems using FE-based parametrization". In: *Structural and Multidisciplinary Optimization* 47.4 (2013), pp. 507–521. ISSN: 1615-147X. DOI: 10.1007/s00158-012-0843-z.

[55] M. Fischer et al. “Design of Lightweight Composite Structures: A Parameter Free Structural Optimization Approach”. In: *Key Engineering Materials* 504-506 (2012), pp. 1391–1396. ISSN: 1662-9795. DOI: 10.4028/www.scientific.net/KEM.504-506.1391.

[56] M. Fischer et al. “Optimization of Nonlinear Structures based on Object-Oriented Parallel Programming”. In: *The Seventh International Conference on Engineering Computational Technology*. Civil-Comp Proceedings. Civil-Comp PressStirlingshire, UK, 2010. DOI: 10.4203/ccp.94.67.

[57] R. Fletcher. “A new approach to variable metric algorithms”. In: *The Computer Journal* 13.3 (1970), pp. 317–322. ISSN: 0010-4620. DOI: 10.1093/comjnl/13.3.317.

[58] R. Fletcher. “Function minimization by conjugate gradients”. In: *The Computer Journal* 7.2 (1964), pp. 149–154. ISSN: 0010-4620. DOI: 10.1093/comjnl/7.2.149.

[59] A. S. Fraser. “Simulation of genetic systems by automatic digital computers vi. epistasis”. In: *Australian Journal of Biological Sciences 13.2* 13.2 (1960), pp. 150–162.

[60] U. Gabbert, T. Nestorović-Trajkov, and H. Köppe. “Finite element-based overall design of controlled smart structures”. In: *Structural Control and Health Monitoring* 13.6 (2006), pp. 1052–1067. ISSN: 15452255. DOI: 10.1002/stc.93.

[61] F. Gandhi, M. Frecker, and A. Nissly. “Design Optimization of a Controllable Camber Rotor Airfoil”. In: *AIAA Journal* 46.1 (2008), pp. 142–153. ISSN: 0001-1452. DOI: 10.2514/1.24476.

[62] D. Gates. *In Person: Fitzgerald’s fix for Boeing 747-8 earns aviation honors*. URL: http://www.seattletimes.com/business/in-person-fitzgeralds-fix-for-boeing-

747-8-earns-aviation-honors/ (visited on 03/23/2015).

[63] M. B. Giles and N. A. Pierce. “An Introduction to the Adjoint Approach to Design”. In: *Flow, Turbulence and Combustion* 65 (2000), pp. 393–415.

[64] D. Goldfarb. “A family of variable-metric methods derived by variational means”. In: *Mathematics of Computation* 24.109 (1970), p. 23. ISSN: 0025-5718. DOI: 10.1090/S0025-5718-1970-0258249-6.

[65] S. D. Guest and S. Pellegrino. “Analytical models for bistable cylindrical shells”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 462.2067 (2006), pp. 839–854. ISSN: 1364-5021. DOI: 10.1098/rspa.2005.1598.

[66] H. Haase. *Bruchlinientheorie von Platten: Grundlagen und Anwedungen*. Düsseldorf: Werner-Verlag, 1962.

[67] R. T. Haftka and Z. Gürdal. *Elements of structural optimization*. 3rd rev. and expanded ed. Vol. v. 11. Solid mechanics and its applications. Dordrecht and Boston: Kluwer Academic Publishers, 1992. ISBN: 9780792315056.

[68] L. Harzheim. *Strukturoptimierung: Grundlagen und Anwendungen*. 1. Aufl. Frankfurt am Main: Deutsch, 2008. ISBN: 978 3 8171 1809 0.

[69] J. Haslinger and R. A. Mäkinen. *Introduction to shape optimization: Theory, approximation, and computation*. Advances in design and control. Philadelphia: SIAM, Society for Industrial and Applied Mathematics, 2003. ISBN: 0898715369.

[70] E. J. Haug, K. K. Choi, and V. Komkov. *Design sensitivity analysis of structural systems*. Vol. v. 177. Mathematics in science and engineering. Orlando: Academic Press, 1986. ISBN: 9780080960005.

[71] M. R. Hestenes. “Multiplier and gradient methods”. In: *Journal of Optimization Theory and Applications* 4.5 (1969), pp. 303–320. ISSN: 0022-3239. DOI: 10.1007/BF00927673.

[72] M. Hojjat, E. Stavropoulou, and K.-U. Bletzinger. “The Vertex Morphing method for node-based shape optimization”. In: *Computer Methods in Applied Mechanics and Engineering* 268 (2014), pp. 494–513. ISSN: 00457825. DOI: 10.1016/j.cma.2013.10.015.

[73] T. J. R. Hughes. *The finite element method: Linear static and dynamic finite element analysis*. Englewood Cliffs and N.J: Prentice-Hall, ©1987. ISBN: 978-0133170177.

[74] T. J. R. Hughes and W. K. Liu. “Nonlinear finite element analysis of shells: Part I. three-dimensional shells”. In: *Computer Methods in Applied Mechanics and Engineering* 26.3 (1981), pp. 331–362. ISSN: 00457825.

[75] M. W. Hyer. “Calculations of the room-temperature shapes of unsymmetric laminates”. In: *Journal of Composite Materials* 15 (1981), pp. 296–310.

[76] A. H. Ibrahim and S. N. Tiwari. “Variational sensitivity analysis and design optimization”. In: *Computers & Fluids* 38.10 (2009), pp. 1887–1894. ISSN: 00457930. DOI: 10.1016/j.compfluid.2009.04.008.

[77] A. Jameson. “Optimum aerodynamic design using control theory”. In: *Computational Fluid Dynamics Review* 3 (1995), pp. 495–528.

[78] K. W. Johanses. *Brudlinieteorier*. Copenhagen: Jul. Gjellerups Forlag, 1943.

[79] R. M. Jones. *Mechanics of composite materials*. 2nd ed. Philadelphia and PA: Taylor & Francis, 1999. ISBN: 978-1560327127.

[80] W. Karush. *Minima of Functions of Several Variables with Inequalities as Side Constraints*. Chicago, 1939.

[81] E. Kebadze, S. D. Guest, and S. Pellegrino. "Bistable prestressed shell structures". In: *International Journal of Solids and Structures* 41.11-12 (2004), pp. 2801–2820. ISSN: 00207683. DOI: 10.1016/j.ijsolstr.2004.01.028.

[82] J. Kennedy and R. Eberhart. "Particle swarm optimization". In: *ICNN'95 - International Conference on Neural Networks*. 27 Nov.-1 Dec. 1995, pp. 1942–1948. DOI: 10.1109/ICNN.1995.488968.

[83] J. F. Kennedy, R. C. Eberhart, and Y. Shi. *Swarm intelligence*. The Morgan Kaufmann series in evolutionary computation. San Francisco: Morgan Kaufmann Publishers, 2001. ISBN: 1-55860-595-9.

[84] S. Kimmich and E. Ramm. "Structural Optimization and Analysis with Program System CARAT". In: *Discretization Methods and Structural Optimization — Procedures and Applications*. Ed. by C. A. Brebbia et al. Vol. 42. Lecture Notes in Engineering. Berlin and Heidelberg: Springer Berlin Heidelberg, 1989, pp. 186–193. ISBN: 978-3-540-50784-0. DOI: 10.1007/978-3-642-83707-4_24.

[85] H. W. Kuhn and A. W. Tucker. "Nonlinear Programming". In: *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability* (1951), pp. 481–492.

[86] T. Lindby and J. L. T. Santos. “Shape optimization of three-dimensional shell structures with the shape parametrization of a CAD system”. In: *Structural Optimization* 18.2-3 (1999), pp. 126–133. ISSN: 0934-4373. DOI: 10.1007/BF01195987.

[87] E. Lindgaard and E. Lund. “Nonlinear buckling optimization of composite structures”. In: *Computer Methods in Applied Mechanics and Engineering* 199.37-40 (2010), pp. 2319–2330. ISSN: 00457825. DOI: 10.1016/j.cma.2010.02.005.

[88] K. H. Lo, R. M. Christensen, and E. M. Wu. “A High-Order Theory of Plate Deformation—Part 2: Laminated Plates”. In: *Journal of Applied Mechanics* 44.4 (1977), p. 669. ISSN: 00218936. DOI: 10.1115/1.3424155.

[89] LosHawlos. URL: https://commons.wikimedia.org/w/index.php?curid=369233 (visited on 05/07/2015).

[90] H. Masching and K.-U. Bletzinger. “Parameter free structural optimization applied to the shape optimization of smart structures”. In: *Finite Elements in Analysis and Design* 111 (2016), pp. 33–45. ISSN: 0168874X. DOI: 10.1016/j.finel.2015.12.008.

[91] H. Masching et al. “Finite Element based Structural Optimization using Object-Oriented Parallel Programming”. In: *Proceedings of the Second International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering*. Ed. by P. Iványi and B.H.V. Topping. Stirlingshire: Civil-Comp Press, 2011.

[92] H. Masching et al. “Parameter Free Structural Optimization of Large Lightweight Composite Structures”. In: *Mechanical response of composites*. Ed. by Raimund Rolfes. Hannover: Univ. Hannover Institute of Structural Analysis, 2011. ISBN: 978-3-00-035855-5.

[93] F. Mattioni et al. “Analysis of thermally induced multistable composites”. In: *International Journal of Solids and Structures* 45.2 (2008), pp. 657–675. ISSN: 00207683. DOI: 10.1016/j.ijsolstr.2007.08.031.

[94] A. Meister and C. Vömel. *Numerik linearer Gleichungssysteme: Eine Einführung in moderne Verfahren; mit MATLAB-Implementierungen von C. Vömel*. 3., überarb. Aufl. Wiesbaden: Vieweg, 2008. ISBN: 978-3-8348-0431-0.

[95] M. Mitchell. *An introduction to genetic algorithms*. 1st MIT Press pbk. ed. Complex adaptive systems. Cambridge and Mass: MIT Press, 1998, c1996. ISBN: 0262631857.

[96] J. Nocedal and S. J. Wright. *Numerical optimization*. 2nd ed. Springer series in operations research and financial engineering. New York: Springer, 2006. ISBN: 978-0-387-30303-1.

[97] N. Olhoff, M. P. Bendsøe, and J. Rasmussen. “On CAD-integrated structural topology and design optimization”. In: *Computer Methods in Applied Mechanics and Engineering* 89.1-3 (1991), pp. 259–279. ISSN: 00457825. DOI: 10.1016/0045-7825(91)90044-7.

[98] N. Olhoff and J. Rasmussen. “Study of inaccuracy in semi-analytical sensitivity analysis — a model problem”. In: *Structural Optimization* 3.4 (1991), pp. 203–213. ISSN: 0934-4373. DOI: 10.1007/BF01744055.

[99] S. Opitz, A. D. Gardner, and K. Kaufmann. “Aerodynamic and structural investigation of an active back-flow flap for dynamic stall control”. In: *CEAS Aeronautical Journal* 5.3 (2014), pp. 279–291. ISSN: 1869-5582. DOI: 10.1007/s13272-014-0106-3.

[100] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. Computer science and applied mathematics. New York: Academic Press, 1970. ISBN: 978-0898714616.

[101] D. A. Pierre. *Optimization theory with applications*. New York: Dover Publications, 1986. ISBN: 978-0486652054.

[102] P. M. Pinsky and K. O. Kim. “A multi-director formulation for elastic—viscoelastic layered shells”. In: *International Journal for Numerical Methods in Engineering* 23.12 (1986), pp. 2213–2244. ISSN: 0029-5981. DOI: 10.1002/nme.1620231206.

[103] O. Pironneau. *Optimal Shape Design for Elliptic Systems*. Springer Series in Computational Physics. Berlin and Heidelberg: Springer Berlin Heidelberg, 1984. ISBN: 978-3-642-87724-7.

[104] A. Pirrera, D. Avitabile, and P. M. Weaver. “Bistable plates for morphing structures: A refined analytical approach with high-order polynomials”. In: *International Journal of Solids and Structures* 47.25-26 (2010), pp. 3412–3425. ISSN: 00207683. DOI: 10.1016/j.ijsolstr.2010.08.019.

[105] A. Pirrera, D. Avitabile, and P. M. Weaver. “On the thermally induced bistability of composite cylindrical shells for morphing structures”. In: *International Journal of Solids and Structures* 49.5 (2012), pp. 685–700. ISSN: 00207683. DOI: 10.1016/j.ijsolstr.2011.11.011.

[106] M. Powell. “A method for nonlinear constraints in minimization problems”. In: *Optimization* (1969), pp. 283–298.

[107] E. Ramm. *Buckling of shells: Proceedings of a state-of-the-art colloquium; Stuttgart, May 6-7, 1982*. Berlin a.o: Springer, 1982. ISBN: 3540117857.

[108] J. N. Reddy. “A Simple Higher-Order Theory for Laminated Composite Plates”. In: *Journal of Applied Mechanics* 51.4 (1984), p. 745. ISSN: 00218936. DOI: `10.1115/1.3167719`.

[109] M. Schenk. *Folded Shell Structures*. Dissertation at the University of Cambridge, 2011.

[110] M. Scherer, R. Denzer, and P. Steinmann. “A fictitious energy approach for shape optimization”. In: *International Journal for Numerical Methods in Engineering* (2009), n/a. ISSN: 00295981. DOI: `10.1002/nme.2764`.

[111] A. Schumacher. *Optimierung mechanischer Strukturen: Grundlagen und industrielle Anwendungen*. 2., aktual. u. erg. Aufl. 2013. SpringerLink : Bücher. Berlin and Heidelberg: Springer Vieweg, 2013. ISBN: 978-3-642-34699-6.

[112] A. Schumacher and R. Hierold. “Parameterized CAD-models for muitidisciplinary optimization processes”. In: *8th AIAA/USAF/-NASA/ISSMO, Symposium on Multidisciplinary Analysis and Optimization* (2000).

[113] K. A. Seffen. “Compliant shell mechanisms”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 370.1965 (2012), pp. 2010–2026. ISSN: 1364-503X. DOI: `10.1098/rsta.2011.0347`.

[114] D. F. Shanno. “Conditioning of quasi-Newton methods for function minimization”. In: *Mathematics of Computation* 24.111 (1970), p. 647. ISSN: 0025-5718. DOI: `10.1090/S0025-5718-1970-0274029-X`.

[115] L. Pereira da Silva, W. Larbi, and J.-F. Deu. “Topology optimization of shunted piezoelectric elements for structural vibration reduction”. In: *Journal of Intelligent Material Systems and Structures* 26.10 (2015), pp. 1219–1235. DOI: `10.1177/1045389X14538533`.

[116] J. C. Simo and M. S. Rifai. “A class of mixed assumed strain methods and the method of incompatible modes”. In: *International Journal for Numerical Methods in Engineering* 29.8 (1990), pp. 1595–1638. ISSN: 0029-5981. DOI: 10.1002/nme.1620290802.

[117] H. Singh, A. Kumar, and G. Singh. “Image Morphing: A Literature Study”. In: *International Journal of Computer Applications Technology and Research* 3.11 (2014), pp. 701–705.

[118] J. Sokolowski and J.-P. Zolesio. *Introduction to Shape Optimization: Shape Sensitivity Analysis*. Vol. 16. Springer series in computational mathematics. Berlin and Heidelberg: Springer Berlin Heidelberg, 1992. ISBN: 978-3-642-58106-9.

[119] P. Sonnabend. URL: https://de.wikipedia.org/wiki/Kompensator_(Rohr)#/media/File:Kompensatoren.jpg (visited on 04/04/2016).

[120] E. D. Sontag. *Mathematical control theory: Deterministic finite dimensional systems*. 2nd ed. Vol. 6. Texts in applied mathematics. New York: Springer, 1998. ISBN: 978-0387984896.

[121] E. Stavropoulou, M. Hojjat, and K.-U. Bletzinger. “In-plane mesh regularization for node-based shape optimization problems”. In: *Computer Methods in Applied Mechanics and Engineering* 275 (2014), pp. 39–54. ISSN: 00457825. DOI: 10.1016/j.cma.2014.02.013.

[122] J. Stegmann and E. Lund. “Discrete material optimization of general composite shell structures”. In: *International Journal for Numerical Methods in Engineering* 62.14 (2005), pp. 2009–2027. ISSN: 00295981. DOI: 10.1002/nme.1259.

[123] G. Strang. *Introduction to linear algebra*. 4th ed. Wellesley and MA: Wellesley-Cambridge Press, 2009. ISBN: 978-0-9802327-1-4.

[124] K. Y. Sze, X. H. Liu, and S. H. Lo. "Popular benchmark problems for geometric nonlinear analysis of shells". In: *Finite Elements in Analysis and Design* 40.11 (2004), pp. 1551–1569. ISSN: 0168874X. DOI: `10.1016/j.finel.2003.11.001`.

[125] R. L. Taylor, P. J. Beresford, and E. L. Wilson. "A non–conforming element for stress analysis". In: *International Journal for Numerical Methods in Engineering* 10 (1976), pp. 1211–1219.

[126] M. Ulbrich and S. Ulbrich. *Nichtlineare Optimierung*. Mathematik Kompakt. Basel: Birkhäuser Verlag, 2012. ISBN: 3034601425.

[127] Universität Stuttgart. *Stuttgart SmartShell*. URL: `http://www.uni-stuttgart.de/hkom/presseservice/pressemitteilungen/2012/022_ultraleichtbau?__locale=en` (visited on 03/23/2015).

[128] Unknown. *System testing for 747 wingtip flutter problem shows a major success*. URL: `http://www.vectorcast.com/news/embedded-software-testing/avionics-aerospace-and-defense-software/system-testing-747-wingtip` (visited on 03/23/2015).

[129] S. Y. Wang, Y. Sun, and R. H. Gallagher. "Sensitivity analysis in shape optimization of contiuum structures". In: *Computers and Structures* 20 (1985), pp. 855–867.

[130] Z. Wang, Y. Cao, and Y. Zhao. "Modeling and optimal design for static shape control of smart reflector using simulated annealing algorithm". In: *Journal of Intelligent Material Systems and Structures* (2015). DOI: `10.1177/1045389X15577650`.

[131] Wikipedia. URL: `https://en.wikipedia.org/wiki/File:Cargolux_747-8F_N5573S_over_Fresno.jpg` (visited on 04/04/2016).

[132] N. Will, R. Littwin, and J. Hegger. *Berechnung von Stahlbetonplatten nach der Bruchlinientheorie*. Fraunhofer IRB Verlag, 2003. ISBN: 978-3816778905.

[133] E. L. Wilson et al. “Incompatible displacement models (isoparametric finite elements in solid and thick shell structural analysis)”. In: *Numerical and computer methods in structural mechanics* (1973), pp. 43–57.

[134] P. Wolfe. “Convergence Conditions for Ascent Methods”. In: *SIAM Review* 11.2 (1969), pp. 226–235. ISSN: 0036-1445. DOI: 10.1137/1011036.

[135] P. Wriggers. *Nonlinear finite element methods*. Berlin: Springer, 2008. ISBN: 3540710000.

[136] G. Yu et al. “CAD-based shape optimisation using adjoint sensitivities”. In: *Computers & Fluids* 46.1 (2011), pp. 512–516. ISSN: 00457930. DOI: 10.1016/j.compfluid.2011.01.043.

[137] O. C. Zienkiewicz and R. L. Taylor. *Finite Element Method: Vol. 2: Solid and Structural Mechanics*. 5th ed. Vol. vol. 2. Finite Element Method Ser. San Diego: Elsevier Science & Technology Books, Sept. 2000. ISBN: 978-0750650557.

[138] O. C. Zienkiewicz and R. L. Taylor. *The finite element method*. 5th ed. Oxford and Boston: Butterworth-Heinemann, 2000. ISBN: 978-0750650496.

Schriftenreihe

In dieser Schriftenreihe sind bisher erschienen

1. F. Koschnik. *Geometrische Lockingeffekte bei Finiten Elementen und ein allgemeines Konzept zu ihrer Vermeidung*, 2004
2. N. Camprubi. *Design and Analysis in Shape Optimization of Shells*, 2004
3. B. Thomée. *Physikalisch nichtlineare Berechnung von Stahlfaserbetonkonstruktionen*, 2005
4. F. Daoud. *Formoptimierung von Freiformschalen - Mathematische Algorithmen und Filtertechniken*, 2005
5. M. Bischoff. *Models and Finite Elements for Thin-walled Structures*, 2005
6. A. Hörmann. *Ermittlung optimaler Stabwerkmodelle auf Basis des Kraftflusses als Anwendung plattformunabhängiger Prozesskopplung*, 2006
7. R. Wüchner. *Mechanik und Numerik der Formfindung und Fluid-Struktur-Interaktion von Membrantragwerken*, 2006
8. F. Jurecka. *Robust Design Optimization Based on Metamodeling Techniques*, 2007

9. J. Linhard. *Numerisch-Mechanische Betrachtung des Entwurfsprozesses von Membrantragwerken*, 2009

10. A. Kupzok. *Modeling the Interaction of Wind and Membrane Structures by Numerical Simulation*, 2009

11. B. Yang. *Modified Particle Swarm Optimizers and their Application to Robust Design and Structural Optimization*, 2009

12. M. Fleischer. *Absicherung der virtuellen Prozesskette für Folgeoperationen in der Umformtechnik*, 2009

13. A. Jrusjrungkiat. *Nonlinear Analysis of Pneumatic Membranes - From Subgrid to Interface*, 2009

14. A. Michalski. *Simulation leichter Flächentragwerke in einer numerisch generierten atmosphärischen Grenzschicht*, 2010

15. M. Firl. *Optimal Shape Design of Shell Structures*, 2010

16. T. Gallinger. *Effiziente Algorithmen zur partitionierten Lösung stark gekoppelter Probleme der Fluid-Struktur-Wechselwirkung*, 2011

17. J. Kiendl. *Isogeometric Analysis and Shape Optimal Design of Shell Structures*, 2011

18. J. Jordan. *Effiziente Simulation großer Mauerwerksstrukturen mit diskreten Rissmodellen*, 2011

19. A. von Boetticher. *Flexible Hangmurenbarrieren: Eine numerische Modellierung des Tragwerks, der Hangmure und der Fluid-Struktur-Interaktion*, 2012

20. R. Schmidt. *Trimming, Mapping and Optimization in Isogeometric Analysis of Shell Structures*, 2013

21. M. Fischer. *Finite Element Based Simulation, Design and Control of Piezoelectric and Lightweight Smart Structures*, 2013

22. F. Dieringer. *Numerical Methods for the Design and Analysis of Tensile Structures*, 2014

23. R. Fisch. *Code Verification of Partitioned FSI Environments for Lightweight Structures*, 2014

24. S. Sicklinger. *Stabilized Co-Simulation of Coupled Problems including Fields and Signals*, 2014

25. M. Hojjat. *Node-based Parameterization for Shape Optimal Design*, 2015

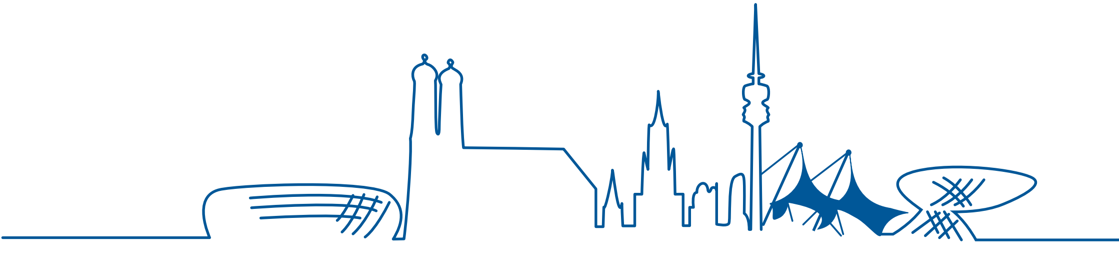
26. U. Israel. *Optimierung in der Fluid-Struktur-Interaktion - Sensitivitätsanalyse für die Formoptimierung auf Grundlage des partitionierten Verfahrens*, 2015

27. E. Stavropoulou. *Sensitivity Analysis and Regularization for Shape Optimization of Coupled Problems*, 2015

28. D. Markus. *Numerical and Experimental Modeling for Shape Optimization of Offshore Structures*, 2015

29. P. Suares. *Design Process for the Optimization of Pressurized Bulkheads as Components of Aircraft Structures*, 2015

30. A. Widhammer. *Variation of Reference Strategy - Generation of Optimized Cutting Patterns for Textile Fabrics*, 2015



ISBN 978-3-943683-37-0