



TECHNISCHE UNIVERSITÄT MÜNCHEN

Ingenieurfacultät Bau Geo Umwelt

Lehrstuhl für Statik

**Vertex Morphing for constrained shape optimization of  
three-dimensional solid structures**

Franz-Josef Ertl

Vollständiger Abdruck der von der Ingenieurfacultät Bau Geo Umwelt der Technischen Universität München zur Erlangung des akademischen Grades 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. Gerhard Müller
3. Prof. Dr.-Ing. Axel Schumacher

Die Dissertation wurde am 18.05.2020 bei der Technischen Universität München eingereicht und durch die Ingenieurfacultät Bau Geo Umwelt am 02.11.2020 angenommen.



Schriftenreihe des Lehrstuhls für Statik  
der Technischen Universität München

Band 45

Franz-Josef Ertl

**Vertex Morphing for constrained shape optimization of  
three-dimensional solid structures**

München 2020

Veröffentlicht durch

Kai-Uwe Bletzinger

Lehrstuhl für Statik

Technische Universität München

Arcisstr. 21

80333 München

Telefon: +49(0)89 289 22422

Telefax: +49(0)89 289 22421

E-Mail: [kub@tum.de](mailto:kub@tum.de)

Internet: [www.st.bgu.tum.de](http://www.st.bgu.tum.de)

ISBN: 978-3-943683-62-2

©Lehrstuhl für Statik, TU München

---

## Abstract

CAO (Computer Aided Optimization) tools and particularly the field of numerical shape optimization offer new possibilities to considerably improve existing designs. However, these tools are still away from being part of a standard design process. This work tries to take a step towards the integration of a shape optimization tool in the design process of turbomachinery components.

In this work a complete shape optimization workflow based on the *Vertex Morphing* method is developed. Generally, *Vertex Morphing* is a parameter-free, gradient-based shape optimization method where the shape is described solely by its discretization providing the widest possible design space. A drawback of this description is the noisy sensitivity field resulting in a non-smooth shape update. Therefore, in addition to the geometry a design control field is introduced on which the mathematical optimization problem is defined. Both fields are related by a linear mapping based on an explicit filter which is consistently incorporated in the theory. In this way, the ill-posed shape optimization problem is overcome and meaningful shapes are obtained.

The extension developed in this work for the optimization of geometrically complex three-dimensional aerospace components comprises the implementation of the adjoint sensitivity analysis method for the relevant response functions. As constraints are immanent in industrial applications, the Gradient Projection method is added in the workflow projecting the function gradient on the subspace tangent of all active constraints to obtain feasible designs. The evolution of the design is achieved by an improved Traction method containing a distance dependent stiffness distribution and an inclusion of invariable areas in the design update further reducing mesh irregularities and mesh dependencies.

The overall design chain is presented and tested on several academic and real size engineering problems. The method shows to be robust and very efficient even for highly sophisticated problems with many design variables, complex geometries and numerous active constraints.

---

## Vorwort

Die vorliegende Arbeit entstand in den Jahren 2015 bis 2020 während meiner Zeit als Doktorand am Lehrstuhl für Statik der Technischen Universität München.

Meinen besonderen Dank möchte ich meinem Doktorvater und Inhaber des Lehrstuhls Herrn Prof. Dr.-Ing. Kai-Uwe Bletzinger aussprechen, der mit seinem breiten Fachwissen auf dem Gebiet der Optimierung mir stets als Ratgeber zur Verfügung stand und mir gleichzeitig alle Freiheiten gewährte eigene Ideen umzusetzen. Durch diese vertrauensvolle Zusammenarbeit ist die vorliegende Arbeit erst möglich geworden.

Herrn Prof. Dr.-Ing. habil. Gerhard Müller und Herrn Prof. Dr.-Ing. Axel Schumacher danke ich herzlich für ihr Interesse an der Arbeit und für die Übernahme des Koreferats. Ebenso danke ich Herrn Prof. Dr.-Ing. Fabian Duddeck für die bereitwillige Übernahme der Leitung der Prüfungskommission.

Bedanken möchte ich mich bei meinen Teamleitern, Dr. Wolfgang Diepolder, Dr. Alexander Buck, Dr. Gregor Kotucha und Dr. Mark Welling. Sie haben mir neben den beruflichen Verpflichtungen stets genügend Freiraum für die wissenschaftliche Arbeit geschaffen. Hervorheben möchte ich auch meinen Abteilungsleiter Dr. Hans-Peter Hackenberg, der aufgrund seines großen Interesses an der Optimierung die Forschung auf diesem Themengebiet immer gefördert hat. Mein besonderer Dank gilt meinem Betreuer Dr. Guido Dhondt, der mich geduldig in das Programm CalculiX eingearbeitet hat und mit seinem breiten Fachwissen in Mechanik und Numerik immer beratend zur Seite stand.

Ebenfalls will ich meiner Familie und besonders meiner Mutter Anna für den gewährten Rückhalt und die Unterstützung danken. Ein ganz besonderer Dank gebührt jedoch meiner lieben Frau Sandra. Sie hat mir immer den Rücken frei gehalten und mich in schwierigen Phasen auf andere Gedanken gebracht. Dazu bin ich ihr zu größtem Dank verpflichtet.

München, im Dezember 2020

Franz-Josef Ertl

---

# Contents

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.1.1	Adjoint sensitivity analysis of structural problems . . . . .	2
1.1.2	Regularization in node-based shape optimization . . . . .	3
1.1.3	Consideration of constraints . . . . .	5
1.2	Objective of the present work . . . . .	6
<b>2</b>	<b>The Vertex Morphing method</b>	<b>9</b>
2.1	Formulation of the optimization problem . . . . .	10
2.2	Sensitivity analysis . . . . .	11
2.2.1	Overview of methods . . . . .	11
2.2.2	Structural adjoint semi-analytical sensitivity analysis . . . . .	15
2.3	Response functions . . . . .	18
2.3.1	Mass . . . . .	19
2.3.2	Stress . . . . .	20
2.4	Geometrical constraints . . . . .	26
2.4.1	Restricting amount of design change . . . . .	27
2.4.2	Minimum or maximum member size . . . . .	30
2.5	Sensitivity weighting . . . . .	32
2.5.1	Dependency of the discretization . . . . .	32
2.5.2	Influence of the element formulation . . . . .	35
2.6	Regularization of shape optimization problems . . . . .	37
2.6.1	Motivation . . . . .	37
2.6.2	Overview of out-of-surface regularization methods . . . . .	43

2.6.3	Consistent out-of-surface regularization . . . . .	45
2.6.4	Overview of mesh quality control methods . . . . .	51
2.6.5	Mesh update method . . . . .	60
2.6.6	Mesh quality preservation . . . . .	65
<b>3</b>	<b>Optimization workflow</b>	<b>69</b>
3.1	Computational framework . . . . .	70
3.2	Optimization algorithm . . . . .	71
3.2.1	Unconstrained optimization . . . . .	72
3.2.2	Aggregation of constraints . . . . .	74
3.3	Design update . . . . .	79
3.4	Structural assessment and sensitivity analysis . . . . .	80
3.5	Implementation aspects . . . . .	81
<b>4</b>	<b>Application of Vertex Morphing</b>	<b>85</b>
4.1	Holes in a plate . . . . .	86
4.1.1	Single hole in a plate . . . . .	86
4.1.2	Two closely-spaced holes in a plate . . . . .	90
4.2	Fillet design . . . . .	94
4.2.1	Reduction of notch stresses . . . . .	95
4.2.2	A filter and step size study . . . . .	97
4.3	Low pressure turbine blade . . . . .	99
4.3.1	Constrained optimization of stresses . . . . .	99
4.3.2	A time consumption study . . . . .	106
4.4	Case shell . . . . .	110
<b>5</b>	<b>Conclusion and outlook</b>	<b>121</b>
<b>A</b>	<b>Transformation of partial derivative of the stress response function</b>	<b>125</b>
<b>B</b>	<b>Efficient computation of projected gradient vector</b>	<b>127</b>
	<b>Bibliography</b>	<b>129</b>



# Introduction

---

## 1.1 Motivation

Especially in aerospace industry the reliability of the different components is of major interest. For this reason, in the past the focus mainly was on the conservation of approved designs based on the gained test experience. But in the last few decades there has also been a great interest on improvement and enhancement of the existing components motivated by economic and ecological reasons. Computer aided optimization tools and particularly the field of numerical shape optimization offer new possibilities to considerably improve these designs. However, optimization software is still away from being part in the standard design process in industry. This work tries to take a step towards the integration of the *Vertex Morphing* method [Ble13, HSB14, Hoj14, SHB14, Sta15, EDB19] as a shape optimization tool in the design process of turbomachinery components.

In every optimization process a key element is the parametrization of the shape. In practice Computer Aided Design (CAD) methods where the design parameters of the CAD model are defined as design variables are commonly used in structure mechanics optimization problems [HCT<sup>+</sup>99, BF84, ELA11, Kie11]. These parameters can be for instance morphing boxes or the control points of the NURBS. The advantage of these methods is the permanent connection be-

tween the discretized analysis model and its CAD representation throughout the optimization. Furthermore, the limited design space often allows for zero-order methods like evolutionary strategies [Ash06] and genetic algorithms [GH88], both having the advantage that no gradient information is required. Nevertheless, the CAD-based optimization suffers from the low number of design variables resulting in a limited design space with the consequence that the optimized design has always the same features as the initial one.

This drawback motivates the use of larger design spaces which node-based parametrizations can provide. The *Vertex Morphing* method is such a parameter-free, gradient-based optimization method. Herein, the shape is described solely by its discretization providing the largest possible design space. The main disadvantage of this description is the noisy sensitivity field resulting in a non-smooth design [MP01, JMP98]. Therefore, in addition to the geometry a control field is introduced on which the mathematical optimization problem is defined. Both fields are related by a linear mapping based on an explicit filter which is consistently incorporated in the theory. In this way, the ill-posed shape optimization problem is overcome and the nodal sensitivities are transformed to meaningful shapes. So far, the *Vertex Morphing* method has been successfully applied in the context of unconstrained optimization of shell structures [Fir10, AFB10] and fluid-structure interaction problems [Hoj14, Sta15].

This thesis continues the aforementioned work and extends the *Vertex Morphing* method for the optimization of highly three-dimensional solid structures and constrained problems in turbomachinery industry. In the sequence the main topics are further introduced.

### **1.1.1 Adjoint sensitivity analysis of structural problems**

In the *Vertex Morphing* method the vertices of the Finite Element mesh are defined as design variables requiring gradient-based optimization algorithms due to their large number. In such optimization processes the evaluation of the gradient information is one of the crucial steps as it mainly drives the evolution

of the shape.

For problems involving many design variables and only few response functions the adjoint sensitivity analysis has shown to be very efficient leading to a computational cost which is in the same order as the solution of the underlying physical problem [JMP98, SL01]. For this reason it is commonly used for the sensitivity analysis of structural [LBT11, FB12, TM94], fluid [Sch10, Pir74] and coupled problems [MAR04, Sta15]. A major goal of this thesis is to implement the adjoint sensitivity analysis in the open source Finite Element code CalculiX which is used for the structural assessment of turbomachinery components. The challenge here is to introduce a new response function for the optimization of local mechanical stresses based on the Kreisselmeier-Steinhauser formulation [KS79] and the necessary constraint functions to obtain feasible designs. Moreover, [KSWB14] has shown that the computed node-based gradient information can not be directly used as it inherently reflects the influence of the discretization. Thus, the work also comprises the derivation of the necessary weighting steps to achieve mesh independent results.

### **1.1.2 Regularization in node-based shape optimization**

In node-based shape optimization regularization methods are essential [BWDC05] and according to [Fir10] these methods can be separated into out-of-plane regularization and in-plane regularization. Both terms reflect the origin of *Vertex Morphing* in the optimization of two-dimensional shell structures. In case of three-dimensional solid structures these terms are substituted by "out-of-surface regularization" and "regularization of the volume mesh", respectively. The latter methods ensure robust and reliable grids during the optimization whereas the out-of-surface regularization methods transform the noisy sensitivity field in a smooth one.

As aforementioned, the computed node-based sensitivities can not be used directly for the optimization since the problem is ill-posed resulting in non-smooth designs. For this reason, several out-of-surface regularization methods

have been proposed which smooth the raw sensitivities. The class of implicit smoothing methods based on the Sobolev gradient penalizes high curvatures of the sensitivity field by the use of an elliptical equation. It is extensively used and validated particularly for aerodynamic shape optimization problems [JV00, MP09, Sch10, Moh97] where it is included in the sensitivity analysis step as pre-conditioner. Explicit filters are based on convolution integrals where the raw sensitivity field is convoluted with a kernel to obtain the smooth gradients. It is applied in a separate computational step after the sensitivity analysis. These methods are commonly used in the context of structural optimization [LBT11, FB12, FWB13].

With the above mentioned methods the noisy sensitivity field can be transformed to meaningful shapes. However, in the course of the design update the Finite Element mesh typically gets distorted which leads to a reduced accuracy of the results or the optimization even aborts before the optimum is reached [Jia06, Sch11, LBT11]. Therefore, in addition to the gradient smoothing a regularization of the volume mesh is required which maintains a sufficient quality of the three-dimensional grid. Till today various methods have been developed which can globally be partitioned in geometrical and mechanical motivated methods. The geometric methods improve the mesh locally since they are based on a local criterion, e.g. Laplacian smoothing [OBB01, LBT11], whereas the mechanical methods improve the mesh globally by an auxiliary mechanical model. The Traction method [SMA09, RFS<sup>+</sup>14] which uses the elastic smoothing properties of the global stiffness matrix or the implicit in-plane regularization method for two-dimensional shell models [SHB14, Sta15] based on the Updated Reference strategy [BR99, LB10, WB05] can be mentioned.

Typically, both methods are applied consecutively: first the sensitivity smoothing and afterwards the regularization of the volume mesh. Alternatively, the shape and mesh regularity can also be combined [Sch11, HSB14]. In this way, the regularization methods are summarized in one step but the complexity of the problem is increased.

In this work, both regularization methods are executed separately. The smooth design update is achieved by the *Vertex Morphing* method which had already been introduced above. The stability of the optimization process with respect to the mesh quality is controlled by an improved Traction method. It is extended by a distance dependent stiffness distribution and an inclusion of invariable areas in the design update.

### 1.1.3 Consideration of constraints

As aforementioned, the *Vertex Morphing* method so far has been applied to unconstrained optimization problems [FB12, AFB10, HSB14, SHB14] using the simple and robust Steepest Descent method [Rao09] for the design update. But typically in real industrial applications constraints are immanent demanding for more sophisticated optimization algorithms as the design space is divided in a feasible and an infeasible region where the constraints are either fulfilled or violated. The combination of constrained optimization algorithms and the *Vertex Morphing* method is a relatively new area of application. The challenge here is to incorporate the very large design space of *Vertex Morphing* in a suitable optimization strategy.

Regarding the algorithms, direct methods are of central importance for the solution of constrained minimization problems which directly solve the problem by searching through the feasible region for the optimal solution [LY08]. Zoutendijk's Method of Feasible Directions [Zou60] and Rosen's Gradient Projection method [Ros60, Ros61] are two commonly used representatives. The latter one which computes the feasible direction by projecting the function gradient on the subspace tangent of all active constraints has been chosen in this work. The decision which constraints are considered to be active is made by an Active Set method [GMW81, LY08, KT61, Lue69]. Herein, the idea is to partition the constraints into an *active* and an *inactive* group based on the indication of the Lagrange multipliers. In the optimization workflow both algorithms are combined and executed in an iterative manner in every optimization iteration.

## 1.2 Objective of the present work

Still, in automotive, aerospace or marine industry most of the problems solved with node-based optimization techniques consist of smooth surfaces with small curvatures and very limited design spaces. This problem description does not apply at all to the applications presented here. This work tries to take a step towards the integration of parameter-free, gradient-based shape optimization methods in the design process of highly three-dimensional turbomachine components. Therefore, the focus in this work is on the extension of the *Vertex Morphing* method for real size applications in this field. More precisely, the semi-analytic adjoint sensitivity analysis containing all the necessary regularization methods has to be implemented in the open source Finite Element program CalculiX for the required response functions. Additionally, a suitable mesh update method including the preservation of a sufficient mesh quality and a constrained optimization method have to be developed and implemented. Finally, all the single modules are merged to a complete optimization framework that can handle complex industrial optimization problems.

The present thesis is organized as follows:

In *Chapter 2* the *Vertex Morphing* method implemented in the free Finite Element code CalculiX in the course of this research project is presented in detail. Kernel of this gradient-based optimization method is the computation of the sensitivities. Here, the gradients are obtained by the semi-analytic adjoint method also containing the algorithms to relieve the gradient information from the influence of the mesh topology and the element formulation. Typically, the node-based gradients suffer from non-smooth gradient information. This deficiency is cured by an explicit filter which is consistently incorporated in the theory of *Vertex Morphing*. Special emphasis is given to the formulation of the objective function *stress* and *mass* being important quantities in aerospace industry and to some geometrical constraints. Having an accurate sensitivity field for the shape update, the three-dimensional mesh can be adopted towards the optimal design. More precisely, the mesh update vector is obtained as the solution of a pseudo linear-

elastic problem with fictitious boundary conditions.

In *Chapter 3* the aforementioned single modules are wrapped up to a complete optimization framework. Additionally, the algorithm for the constrained aggregation based on the Gradient Projection method is presented. As constraints are immanent in industrial applications, e.g. the maximum admissible mass of a component, the optimizer has to determine feasible designs even for highly constrained problems. Emphasis is also given to some implementation aspects.

*Chapter 4* introduces the various applications of the method. The selection of the test cases is such that each of them challenges the optimization process differently. The chapter contains academic examples to demonstrate accordance with the analytically derived solution up to geometrically complex real-size industrial applications with many design variables and highly constrained design spaces.

At the end, the overall summary as well as suggestions for potential future work are addressed in *Chapter 5*.





# The Vertex Morphing method

---

In this chapter the components for a parameter-free, gradient-based shape optimization process based on the Vertex Morphing method [Ble13, HSB14, EDB19] are presented. The method is capable to optimize three-dimensional solid structures of real-size industrial applications. The chapter is organized as follows: Section 2.1 introduces the general mathematical optimization problem. In section 2.2 the sensitivity analysis which is the kernel of the gradient-based optimization is deduced in detail. In section 2.3 and 2.4 important response functions and geometrical constraints for the optimization of parts from aerospace industry are introduced. The sensitivity weighting curing the influence of the discretization on the gradient information is addressed in section 2.5. At the end of the chapter (section 2.6) the regularization methods necessary to obtain meaningful designs and to prevent a deterioration of the mesh during the optimization are presented.

## 2.1 Formulation of the optimization problem

The optimization problems in mechanical engineering are typically formulated as a mathematical optimization problem. The following discrete problem is considered

$$\begin{aligned}
 & \text{Minimize} && J(\mathbf{s}, \mathbf{u}(\mathbf{s})), && \mathbf{s} \in \mathbb{R}^n \\
 & \text{such that} && G_i(\mathbf{s}, \mathbf{u}(\mathbf{s})) < 0, && i = \{1, \dots, n^G\} \\
 & && H_j(\mathbf{s}, \mathbf{u}(\mathbf{s})) = 0, && j = \{1, \dots, n^H\} \\
 & && \mathbf{s}_l < \mathbf{s} < \mathbf{s}_u
 \end{aligned} \tag{2.1}$$

with the objective function  $J$ , the inequality constraints  $G$ , the equality constraints  $H$ , the design variables  $\mathbf{s}$  and the state variables  $\mathbf{u}$  which are typically the displacements in structure mechanics. The design variables are bounded by the lower constraint  $s_l$  and the upper constraint  $s_u$ , respectively.

By the choice of the type of design variables the basic properties of the optimization problem are characterized. For example in sizing optimization tasks cross section parameters are defined as design variables [LMGM05]. In topology optimization the material density of the Finite Element is considered as design variable [BS04, INR07], whereas in shape optimization problems the CAD representation [HCT<sup>+</sup>99] or the nodes of the Finite Element mesh [FB12] are parameterized. The size  $n$  of the design space  $\mathbb{R}^n$  specifies the number of independent variables which influences the choice of the potential optimization algorithm. If a high number of variables is involved, gradient-based algorithms are very efficient. Zero-order methods are limited to problems with a low number of design variables.

The objective function  $J$  is the measure for the quality of the current design. In most of the problems in structure mechanics one is interested in the improvement of properties like stress, displacement, frequency or mass of a component. The feasible domain can be restricted by  $n^G$  inequality constraint and  $n^H$  equality constraint. During an optimization an inequality constraint can become active, inactive or redundant. Equality constraints are either active or redundant. The

set of all active inequality and equality constraints is denoted as the set of active constraints [Fir10].

## 2.2 Sensitivity analysis

### 2.2.1 Overview of methods

As a general definition, sensitivity analysis quantifies the impact of a variation of the input parameters of a model on its output parameters. Therefore, the sensitivity analysis plays an important role in several areas of application. Among these, inverse and identification studies, error estimation, reliability analysis and numerical optimization can be mentioned. In the field of gradient-based optimization the sensitivities of the objective and the constraint functions are utilized to find a feasible local optimum.

In static problems the response function  $J$  depends on the design variables  $\mathbf{s}$  and on the state variables  $\mathbf{u}$ . Again, the state variables  $\mathbf{u}$  depend on the design variables  $\mathbf{s}$ , which can be expressed as  $J(\mathbf{s}, \mathbf{u}(\mathbf{s}))$ . The discretized governing system of equations of the underlying mechanical problem also depends on  $\mathbf{s}$  and  $\mathbf{u}$ , i.e.  $R(\mathbf{s}, \mathbf{u}(\mathbf{s})) = 0$ . Herein, the vector of the design variables  $\mathbf{s} = [s_1, \dots, s_n]$  is of size  $n$  and the vector of the state variables  $\mathbf{u} = [u_1, \dots, u_m]$  is of size  $m$ , respectively.

Generally, the goal of the sensitivity analysis is to compute the total differential  $dJ/ds$ . In the following, the approaches to assess this derivation are only deduced for the response function  $J$  independent of any gradient-based algorithm. However, many algorithms for constrained problems also require the gradient information of the inequality and equality constraints. The sensitivities for these quantities are computed in exactly the same way as for the response function and are therefore not mentioned separately.

In many cases a simple approach is to compute the total differential with a global finite difference step. The most popular ones are the backward and forward

differences. The latter one is defined as

$$\frac{dJ}{ds_i} = \frac{J(s_i + \Delta s_i) - J(s_i)}{\Delta s_i} + o(\Delta s_i) \quad (2.2)$$

derived from a Taylor series expansion. The first term is the finite difference step and the second one is the truncation error. With a decreasing step size  $\Delta s_i$  the truncation error decreases too. However, a small step size  $\Delta s_i$  amplifies the round-off errors arising from the subtraction of the floating point numbers  $J(s_i + \Delta s_i)$  and  $J(s_i)$ , called the condition error. Consequently, there is an optimal step size for which the error is minimized. Generally, this optimal step size has to be assessed in preparation of the sensitivity analysis. Regarding the effort, the calculation of  $n$  sensitivities with equation (2.2) leads to  $n + 1$  evaluations of the response function  $J$  which typically requires  $n + 1$  solutions of the system of equations  $R(\mathbf{s}, \mathbf{u}(\mathbf{s})) = 0$ . It is obvious that for an increasing number of design variables the computational time quickly becomes prohibitive. Though much effort has been spent to overcome this drawback, e.g. [AGH01] and [KL97], this is the main reason why a sensitivity analysis with global finite differences is not very common in practice even if the implementation is straightforward. More detailed information about this method can be found in [CK06, HA89, KHK05].

A more sophisticated approach is the assessment of the sensitivities by analytical methods. A common classification of the available methods is by the sequence of the discretization and differentiation step. In the discrete approach at first the governing equations are discretized and then differentiated. Here, a deeper understanding of the Finite Element program and access to the source code is compulsory. In a first step the total derivative of the response function with respect to the design variables has to be formulated applying the chain rule of differentiation

$$\frac{dJ}{ds} = \frac{\partial J}{\partial \mathbf{s}} + \frac{\partial J}{\partial \mathbf{u}} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{s}}. \quad (2.3)$$

The terms containing the response function  $J$  can be solved analytically or by means of finite differences. More demanding is the term  $\partial \mathbf{u} / \partial \mathbf{s}$  which is the derivation of the state variables  $\mathbf{u}$  with respect to the design variables  $\mathbf{s}$ . Instead of computing this expression directly, it is assessed by applying the chain rule of

differentiation to the state equation  $\mathbf{R}(\mathbf{s}, \mathbf{u}(\mathbf{s})) = \mathbf{0}$

$$\frac{d\mathbf{R}}{ds} = \frac{\partial \mathbf{R}}{\partial \mathbf{s}} + \frac{\partial \mathbf{R}}{\partial \mathbf{u}} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{s}} = \mathbf{0}. \quad (2.4)$$

By substituting equation (2.4) in equation (2.3), the derivation of the state variables can be replaced

$$\frac{dJ}{ds} = \frac{\partial J}{\partial \mathbf{s}} + \frac{\partial J}{\partial \mathbf{u}} \cdot \frac{\partial \mathbf{R}^{-1}}{\partial \mathbf{u}} \cdot \frac{\partial \mathbf{R}}{\partial \mathbf{s}}. \quad (2.5)$$

Typically, the derivation of the governing equations  $\mathbf{R}$  are the most time consuming parts in equation (2.5). Therefore, it is worth to spent some effort in an efficient computation of these terms. In dependency of the sequence of the evaluation of the particular terms two methods can be distinguished.

The direct approach carries out the evaluation from right to left, which means that first the term

$$\frac{\partial \mathbf{R}^{-1}}{\partial \mathbf{u}} \cdot \frac{\partial \mathbf{R}}{\partial \mathbf{s}} \quad (2.6)$$

is determined by solving the following system of equations

$$\frac{\partial \mathbf{R}}{\partial \mathbf{u}} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{s}} = -\frac{\partial \mathbf{R}}{\partial \mathbf{s}}. \quad (2.7)$$

The right hand side consists of  $n$  columns which means that the system of equations has to be solved  $n$  times. The term  $\partial \mathbf{u} / \partial \mathbf{s}$  can be inserted in equation (2.5) which then requires  $n$  vector multiplications of size  $m$ .

In the adjoint approach, the evaluation starts from the left hand side leading to the following system of equations:

$$\frac{\partial \mathbf{R}^T}{\partial \mathbf{u}} \cdot \lambda = -\frac{\partial J^T}{\partial \mathbf{u}}. \quad (2.8)$$

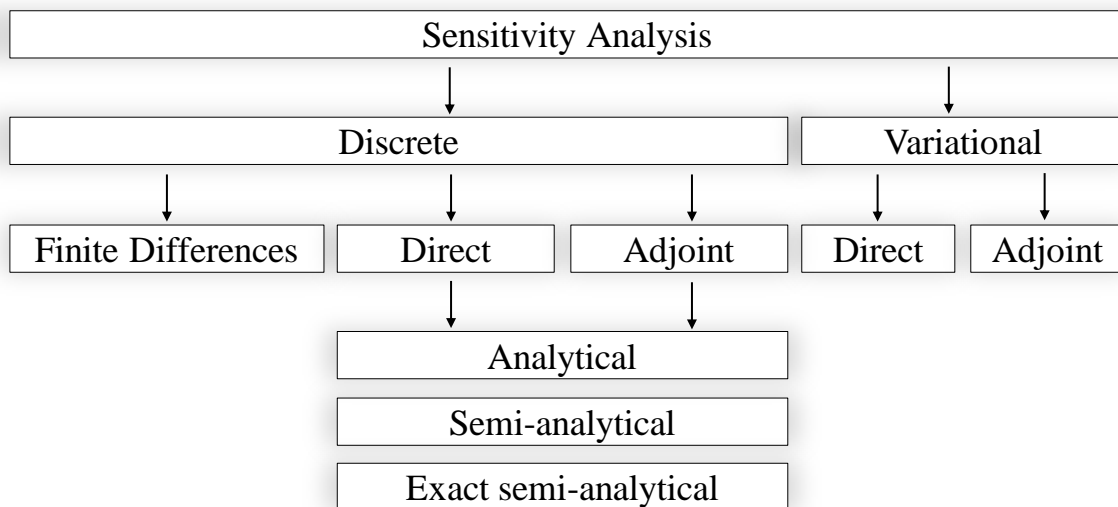
The vector  $\lambda$  is called the adjoint variable. The number of right-hand sides depends on the number of response functions. Hence, this system of equations has to be solved as many times as response functions are defined. At the end the

adjoint variable is substituted back in equation (2.5)

$$\frac{dJ}{ds} = \frac{\partial J}{\partial s} + \lambda^T \cdot \frac{\partial \mathbf{R}}{\partial s}. \quad (2.9)$$

In this equation  $n$  vector multiplications of size  $m$  have to be executed.

From the discussions above it is clear, that the choice between direct and adjoint sensitivity analysis depends on the number of design variables and response functions. In small problems with only a few design variables but many response functions involved in the optimization the discrete approach might be preferred. Otherwise the adjoint method is more efficient. But typically in parameter-free shape optimization, many design variables and only a few response functions are defined. For this type of problems the adjoint sensitivity analysis is the method of choice. Therefore, in this work, the adjoint approach is preferred too.



**Figure 2.1:** Overview of methods for sensitivity analysis

According to figure 2.1 several options exist for the computation of the partial derivatives in the discrete adjoint approach. The sensitivities in the analytical approach are based on exact analytical differentiation of element quantities. It is clear, that this requires a deep knowledge of the element formulation being the main reason why in practice the semi-analytical approach is favored. Here, the sensitivities are approximated at element level with finite differences making

the approach independent from the element formulation. To sum it up, the semi-analytical approach combines ease of implementation with computational efficiency. But as all finite difference steps, however, this approach exhibits truncation and condition errors too, cf. [BH90, PCR89]. To eliminate these accuracy problems, the group of exact semi-analytical methods has been developed. A review of proposed methods for the elimination of errors in semi-analytical design sensitivities can be found in [BFD08, WCB15, LO94, BK00].

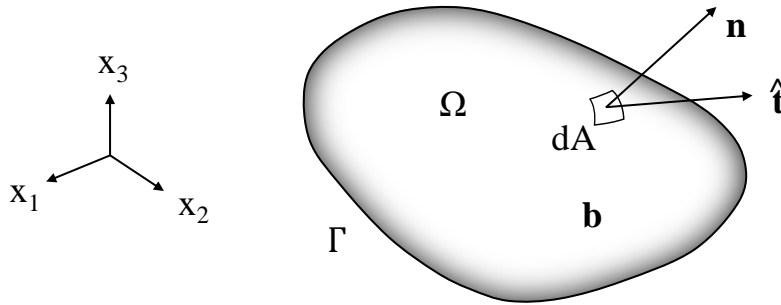
In the variational approach the governing equations are first differentiated and then discretized. In case the source code of commercial structural analysis programs is not accessible or very complex and requires intimate knowledge these kind of methods are applied. Herein, the sensitivity analysis is based on pre- and post-processing operations without any modifications of the original code. At first the continuous formulation of the governing equations are derived, analytically differentiated and finally solved numerically. The goal of this approach is to formulate a pseudo-load vector applied at the right-hand side of the standard structural problem to obtain the state derivative  $\partial \mathbf{u} / \partial s$  at the end, cf. [Dao05]. The main drawback of these kind of methods is that the pseudo load vector has to be derived and implemented for every mechanical problem separately which makes it inflexible. The focus of this work is on discretized approaches using the Finite Element method. Therefore, this type of sensitivity analysis method is only mentioned for the sake of completeness and will not further discussed. For more details, the interested reader is referred to [HG99] and [CK06].

A comprehensive review of the methods used for sensitivity analysis of structural problems can be found in [KHK05, TM94, HA89] and the references therein.

### **2.2.2 Structural adjoint semi-analytical sensitivity analysis**

The discrete sensitivities are based on the discretization of the differential equations of the underlying mechanical problem. In case of the proposed structural shape optimization method these are the equilibrium equations for the linear elastic structure. It states that the change of body momentum is equal to the sum

of all forces acting on this body.



**Figure 2.2:** State of body in equilibrium

For the derivation of the equilibrium conditions let  $\Omega$  be a closed volume bounded by a smooth surface  $\Gamma$ , cf. figure 2.2. Furthermore, a unit normal to the surface  $\mathbf{n}$  pointing outwards is considered. On this body acts a body force  $b_i$  defined on  $\Omega$  and a traction  $\hat{t}_i$  on each boundary surface  $\Gamma_t \subset \Gamma$ . On the boundary surface  $\Gamma_u \subset \Gamma$  and  $\Gamma_u \cap \Gamma_t$  a displacement  $\hat{u}_i$  is prescribed. The goal is to determine the displacement  $u_i$  of the body under the prescribed boundary conditions.

$$\begin{aligned} \frac{\partial \sigma_{ij}}{\partial x_j} + b_i &= 0 \quad \text{in } \Omega \\ u_i &= \hat{u}_i \quad \text{on } \Gamma_u \\ \sigma_{ij} \cdot n_j &= \hat{t}_i \quad \text{on } \Gamma_t \end{aligned} \quad (2.10)$$

The set of equations in (2.10) represents the strong form of the described boundary-value problem which is derived in detail in the textbooks [Hug00, Tim51, Dho04]. The stresses  $\sigma_{ij}$  and the displacements  $u_i$  are connected through the constitutive equation

$$\sigma_{ij} = E_{ijkl} \cdot \epsilon_{kl} \quad (2.11)$$

with  $E_{ijkl}$  being the fourth order elasticity tensor and  $\epsilon_{kl}$  the infinitesimal strain tensor. The strains are derived from the displacements with respect to the spatial coordinates

$$\epsilon_{kl} = \frac{1}{2} \cdot \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right). \quad (2.12)$$

The discretization of equation (2.10) leads to an algebraic system of equations



with the discrete nodal displacements  $\mathbf{u}$  as unknowns

$$\mathbf{R}(\mathbf{s}, \mathbf{u}) = \mathbf{K}(\mathbf{s}, \mathbf{u}(\mathbf{s})) \cdot \mathbf{u}(\mathbf{s}) - \mathbf{f}(\mathbf{s}) = \mathbf{0}. \quad (2.13)$$

The matrix  $\mathbf{K}$  is the stiffness matrix of the problem depending on the design variables  $\mathbf{s}$  and in the geometrically nonlinear case on the displacements  $\mathbf{u}$  too. The vector  $\mathbf{f}$  is called the external load vector which also depends on the design variables.

In structural shape optimization, typically, one is interested in quantities like the mass of a structure, the eigenfrequency, the strain energy, the stresses or displacements at a certain position, cf. section 2.3. These response functions depend directly from the design variables and often indirect from the displacements,

$$J = J(\mathbf{s}, \mathbf{u}(\mathbf{s})) \quad (2.14)$$

In case of structural shape optimization, equation (2.8) reads

$$\mathbf{K}^T \cdot \boldsymbol{\lambda} = -\frac{\partial J^T}{\partial \mathbf{u}}. \quad (2.15)$$

After solving this equation with respect to the adjoint variable  $\boldsymbol{\lambda}$ , the total gradient of the response function can be expressed as

$$\frac{dJ}{d\mathbf{s}} = \frac{\partial J}{\partial \mathbf{s}} + \boldsymbol{\lambda}^T \cdot \mathbf{f}^*. \quad (2.16)$$

Herein,  $\mathbf{f}^*$  is, due to its similarity with the external load vector, the pseudo load "vector" which is defined as

$$\mathbf{f}^* = \frac{\partial \mathbf{K}}{\partial \mathbf{s}} \cdot \mathbf{u} - \frac{\partial \mathbf{f}}{\partial \mathbf{s}}. \quad (2.17)$$

Each column of the matrix represents a pseudo load case defined for each design variable  $s_i$ ,  $i = 1, \dots, n$  [Sta15]. Since the adjoint semi-analytical analysis is applied, all partial derivatives are computed by means of finite differences. The

terms  $\partial J/\partial \mathbf{s}$  and  $\partial J/\partial \mathbf{u}$  are approximated by

$$\frac{dJ}{ds_i} \approx \frac{J(\mathbf{u}, s_i + \Delta s_i) - J(\mathbf{u}, s_i)}{\Delta s_i} \quad (2.18)$$

and

$$\frac{dJ}{du_i} \approx \frac{J(u_i + \Delta u_i, \mathbf{s}) - J(u_i, \mathbf{s})}{\Delta u_i}, \quad (2.19)$$

respectively. The term  $\partial \mathbf{K}/\partial s_i$  requires the assembly of the stiffness matrix for every design variable. In the semi-analytical sensitivity analysis this costly assembly is avoided as the derivatives are computed on element level with finite-differences,

$$\frac{\partial \mathbf{K}}{\partial s_i} \approx \frac{\mathbf{K}(\mathbf{u}, s_i + \Delta s_i) - \mathbf{K}(\mathbf{u}, s_i)}{\Delta s_i}. \quad (2.20)$$

Herein, the main advantage is that the cumbersome element specific derivation of the stiffness matrix does not have to be implemented. Nevertheless, the drawback of this method is the reduced accuracy due to the finite difference step as truncation errors are inherent, cf. [BH90, PCR89]. To identify the erroneous shape design sensitivities, [CO93] introduced the rigid-body motion test for analysis and detection of possible errors in the derivatives. The exact semi-analytical sensitivity analysis overcomes this deficiency. In [BFD08, LO94, BK00] the accuracy is increased by correction factors based on product spaces of rigid body vectors. In case of geometric nonlinear shape optimization [WCB15] proposes a correction based on the product spaces of two sets of zero eigenvectors.

## 2.3 Response functions

Objective and constraint functions are generally denoted as response functions as they typically provide a characteristic property of the design in a scalar value. They depend on the design variables  $\mathbf{s}$  and the state variables  $\mathbf{u}$ . For an application in a gradient-based optimization these functions necessarily have to be differentiable to be able to compute the first order gradient. The algorithms described in this work only consider a single objective function to be optimized. In case more than one objective function is present, multiobjective optimization

algorithms have to be taken into account and the interested reader is referred to [EKO90] and the references therein. In the following, the mass and the stress response function as well as their first order derivatives are described in detail being two important quantities for the optimization of turbomachinery components.

### 2.3.1 Mass

In many structural optimization problems the mass of a component is an important quantity. For example in aerospace industry the mass is directly linked to the fuel consumption which again has an impact on the operating costs of the aircraft. Thus, a light weight design of the different components is of major interest.

In the discretized Finite Element model the total mass  $M$  of a component is determined by the summation of the mass  $m_i$  over the considered elements  $i$  with  $i = 1, \dots, n_{elems}$ . The mass  $m_i$  itself is defined as the product of the volume  $V_i$  of an finite element times its material density  $\rho_i$ ,

$$M = \sum_{i=1}^{n_{elems}} m_i = \sum_{i=1}^{n_{elems}} V_i \cdot \rho_i. \quad (2.21)$$

The number of elements  $n_{elems}$  evaluated in the response function can vary between the whole structure or just a subset, depending from the the underlying optimization problem.

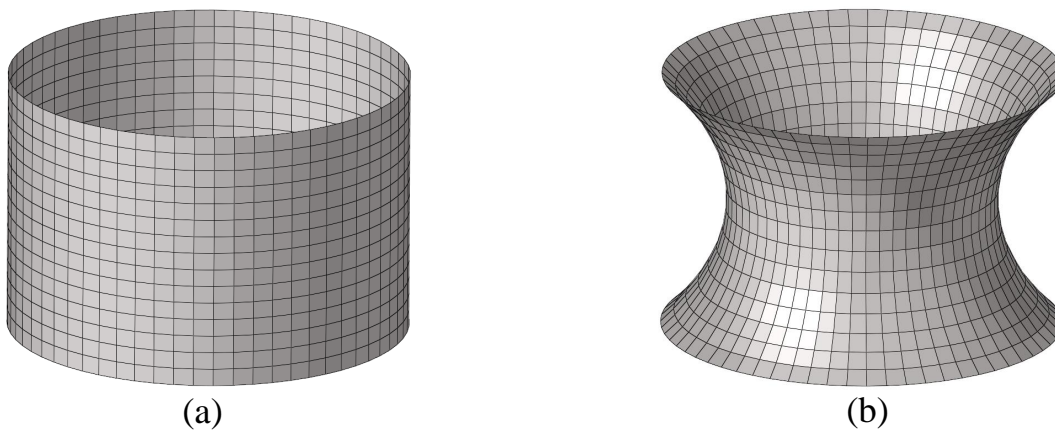
Clearly, the mass  $M$  is only a function of the design but not a function of the state variables and consequently the response function only depends on the design variables,

$$J(\mathbf{s}) = M(\mathbf{s}). \quad (2.22)$$

Thus, the first order derivative of the mass with respect to the design variables form equation (2.3) reduces to

$$\frac{dJ}{ds_i} = \frac{\partial J}{\partial s_i}. \quad (2.23)$$

In an optimization the mass response function can either be used as a constraint or as an objective. The latter one only make sense if further constraints or variable bounds are defined otherwise the optimization converges to trivial solutions, e.g. zero volume designs. One exception is in combination with shell or membrane structures as these elements have per definition a constant thickness. This permits the investigation of minimal surface problems. Figure 2.3(b) shows the famous catenoid from Leonard Euler discovered in 1740. This minimal surface consists of two coaxial rings which are connected by a surface generated from a rotated catenoid around the center axis of the two rings. It is the only minimal surface witch is rotationally symmetric. The initial shape of this catenoid has a height to radius ratio of 1.30, see figure 2.3(a). It can be demonstrated by analytics and experiments that this minimal surface is obtained up to a ratio of 1.32548 [Lin09]. Above this limit, meaning that the distance between the two rings is increased, the catenoid degenerates to two disks at the top and the bottom through a discontinuous transition [BPBA99].



**Figure 2.3:** Catenoid minimum surface, initial shape (a) and final shape (b)

### 2.3.2 Stress

In many of the problems in structural optimization one is interested in the improvement of the stresses as they are directly linked to the life and load capability of a component. However, due to the local nature of the stresses, many positions have to be taken into account. The most obvious approach is to

consider only the maximum stress in the response function with the formula

$$\sigma_{max} = \max_{i=1\dots n_{nodes}} (\sigma_i), \quad (2.24)$$

where  $n_{nodes}$  is the number of nodes taken into account [ALDDR10]. Clearly, this formulation has the advantage that only one node is considered in the optimization problem with the drawback that this function is not differentiable. The gradients of the response function only reflect the influence of the design variables on the worst position. The design update will thus reduce the current largest stress, but the stresses at other locations are not considered and can increase. Due to this alternating positions of the maximum stress the gradients can change significantly from iteration to iteration. The parameter-free optimization even reinforces this effect as very local design changes can be realized [AFB10]. Consequently, if the maximum stress measure is used as response function this discontinuity causes severe problems. More iterations are typically necessary or the algorithm does not converge to the local optimum at all.

Therefore, it is essential to replace the maximum function by a formulation which aggregates all nodal stress values into one continuous scalar function. Every local stress position accounts proportional to its value to the function. This means that the design update also considers the change of the stress value at these positions. In the literature the Kreisselmeier-Steinhauser (KS) function [KS79] and the p-norm are commonly used. Le et al. [LBT11] implemented the p-norm in the context of a parameter-free shape optimization process and demonstrated its capabilities on several academic test cases. In [LCAT<sup>+</sup>17] and [PTB<sup>+</sup>17] the p-norm is successfully integrated in the framework of a combined shape and topology optimization process. Qi et al. [QL10] showed the equivalence between the p-norm and the KS-function and pointed out that the KS-function is even more adaptive to optimization problems. In [LSS15] the nondifferentiability of the maximum stress formulation is avoided by the use of the KS-function. There, the shape of stiffeners on thin-walled structures is optimized to minimize the stress concentration. Poon et al. [PM07] used the KS-function to aggregate all constraints in one closed function for the optimization of a wing structure. If

many constraints are involved the KS-function typically leads to conservative results. This effect can be avoided by updating the aggregation parameters during the optimization leading to more accurate results. Based on these references the KS-function is preferred which is in terms of the stress response function formulated as

$$\sigma_{KS} = \frac{1}{\rho} \cdot \ln \left( \sum_{i=1}^{n_{nodes}} e^{\rho \cdot \frac{\sigma_i}{\bar{\sigma}}} \right) = \frac{1}{\rho} \cdot \ln(\mathcal{H}). \quad (2.25)$$

Herein,  $\sum_{i=1}^{n_{nodes}}$  defines the summation over all nodes within the response function with  $\sigma_i$  being the corresponding nodal stress value. The stresses are scaled by a reference stress  $\bar{\sigma}$  which has a major influence on the behavior of the KS-function. If  $\bar{\sigma}$  is small compared to the highest stress value in the structure the function tends to infinity. On the other hand, if the fraction  $\sigma_i / \bar{\sigma}$  is very small the highest stress values are not emphasized and the function becomes useless. Therefore,  $\bar{\sigma}$  should be in the range of the highest stress value in the response function. The parameter  $\rho$  is a scalar value that determines the importance of the highest stress value in the set. For large  $\rho$  the influence of the KS-function is limited to the neighborhood of the highest stress. For lower values of  $\rho$  the function spreads more even over the structure. For a deeper understanding about the aggregation parameters  $\bar{\sigma}$  and  $\rho$ , Arnout et al. [AFB10] as well as Poon et al. [PM07] analyzed their influence on the optimization result in detail. Concluding, it is important to note that the KS-function, in contrast to the maximum stress function  $\sigma_{max}$ , does not have any physical meaning.

Regarding the stress measure, scalar ones like the von Mises hypothesis, the Tresca hypothesis or the Rankine hypothesis are often better suited than the tensor formulation of the stress. Subsequently, the von Mises stresses will be used for the evaluation of the response function. In three dimensional continuum the von Mises stresses are calculated by

$$\sigma_{vM} = \sqrt{\frac{1}{2} [(\sigma_I - \sigma_{II})^2 + (\sigma_{II} - \sigma_{III})^2 + (\sigma_{III} - \sigma_I)^2]} \quad (2.26)$$

with the principal stresses  $\sigma_I$ ,  $\sigma_{II}$  and  $\sigma_{III}$ .

The stress at an arbitrary nodal position is a function of the design variables and the state variables

$$J(\mathbf{s}, \mathbf{u}) = \sigma_{KS}(\mathbf{s}, \mathbf{u}). \quad (2.27)$$

Therefore, the chain rule of differentiation has to be applied to obtain the following expression

$$\frac{dJ(\mathbf{s}, \mathbf{u})}{d\mathbf{s}} = \frac{\partial \sigma_{KS}}{\partial \mathbf{s}} + \frac{\partial \sigma_{KS}}{\partial \mathbf{u}} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{s}}. \quad (2.28)$$

where  $\partial \mathbf{u} / \partial \mathbf{s}$  denotes the state derivative introduced in section 2.2.1 which is independent from the considered response function. From equation (2.28) it can be concluded that the KS-function has to be derived with respect to the design variables  $\mathbf{s}$  and the displacement field  $\mathbf{u}$ . The derivation of the KS-function follows exactly the same way for the design variables as well as the displacements. Therefore only the first partial derivative  $\partial \sigma_{KS} / \partial \mathbf{s}$  is deduced in detail.

Characteristic for the semi-analytical sensitivity analysis is the approximation of the partial derivatives with a finite difference step

$$\frac{\partial \sigma_{KS}}{\partial s_i} \approx \frac{\frac{1}{\rho} \cdot \ln(\mathcal{H} + \Delta \mathcal{H}) - \frac{1}{\rho} \cdot \ln(\mathcal{H})}{\Delta s_i}. \quad (2.29)$$

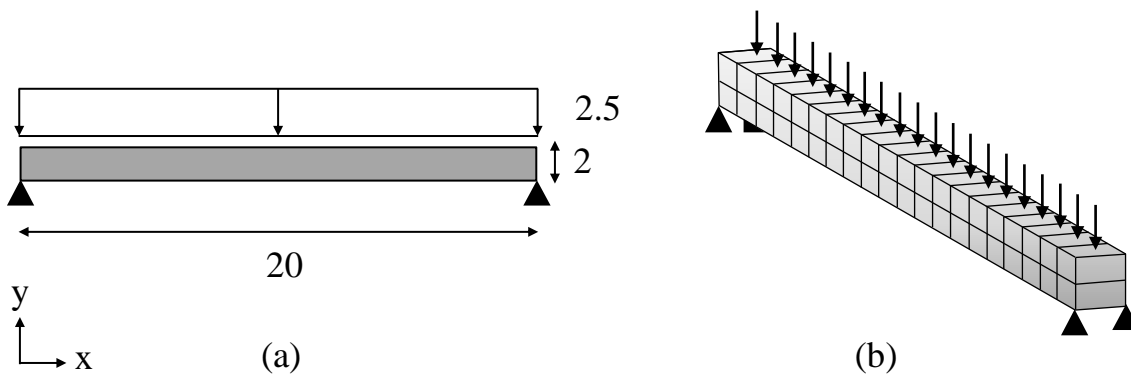
After several transformations, including two linearizations based on a Taylor series expansion, equation (2.29) can be rewritten in the following form

$$\frac{\partial \sigma_{KS}}{\partial s_i} \approx \frac{1}{\bar{\sigma} \cdot \Delta s_i \cdot \mathcal{H}} \cdot \left( \sum_{i=1}^{n_{nodes}} e^{\rho \cdot \frac{\sigma_i}{\bar{\sigma}}} \cdot \Delta \sigma_i \right). \quad (2.30)$$

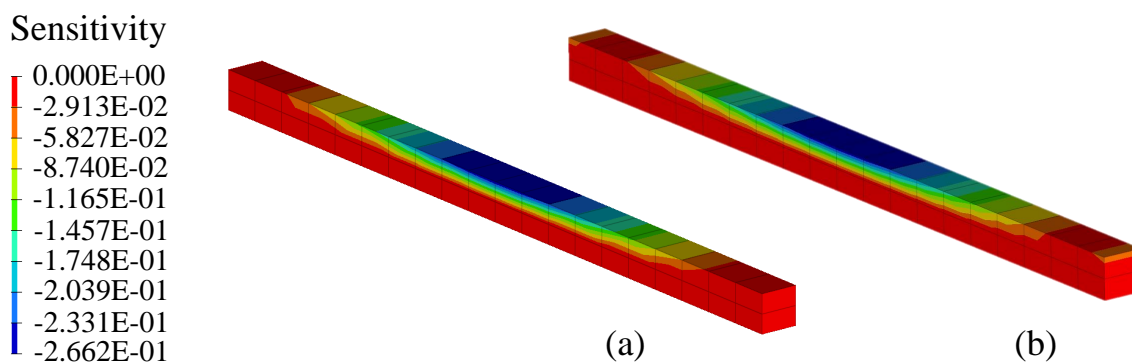
In this expression, except of  $\Delta \sigma_i$  all terms are already available from the computation of the scalar value of the response function. The term  $\Delta \sigma_i$  is received by subtracting the nominal von Mises stress  $\sigma_i$  at node  $i$  from the perturbed one at node  $i$ . The transformation from equation (2.29) to equation (2.30) is described in detail in appendix B.

For the verification of the implemented adjoint sensitivity analysis of the stress response function, a small academic example is evaluated and compared with

the sensitivities gained from a global finite differences step. The underlying mechanical problem is a beam in bending, simply supported at both ends under a constant distributed load of 10 MPa defined along the whole beam. The length of the beam is 20 mm and its cross section is  $1 \times 1$  mm. The problem setup is shown in figure 2.4(a). A Young's modulus of  $2.1 \times 10^5$  MPa, a Poisson ratio of 0.3 and a density of  $7.85 \times 10^{-9}$  t/mm<sup>3</sup> are the corresponding material properties. The geometry is discretized with two rows of 20 linear hexahedral elements, cf. figure 2.4(b). The design variables are defined at the top of the structure. Furthermore, the response function is evaluated for every design variable with  $\rho = 2$  and  $\bar{\sigma} = 200$  MPa as aggregation parameters.



**Figure 2.4:** Beam in bending, mechanical model (a) and Finite Element discretization (b)

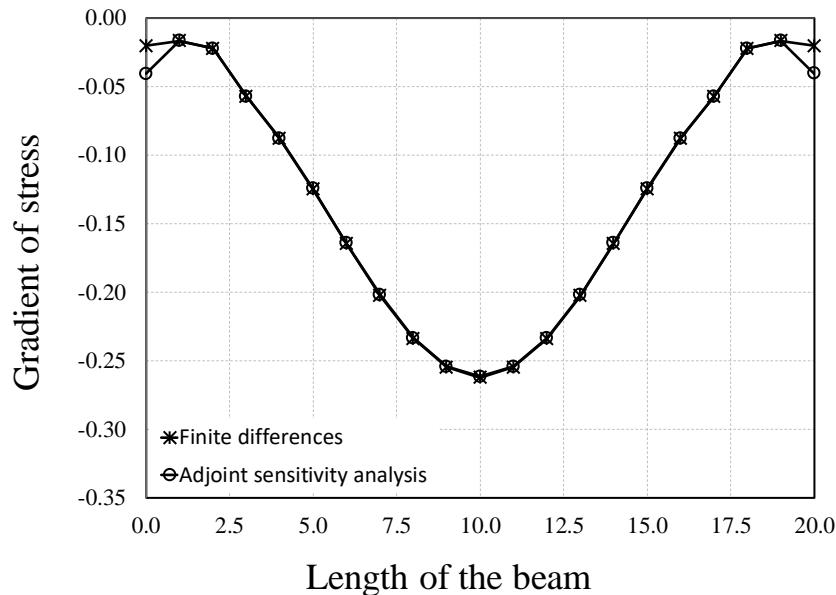


**Figure 2.5:** Comparison of sensitivity field of stress response function computed with finite differences (a) and the adjoint method (b). For the purpose of a better comparison, the length of the gradient vectors has been normalized.



The contour plot in figure 2.5(b) shows the adjoint sensitivity field. It can be seen that all nodal sensitivities are strictly negative with the highest absolute value occurring in the middle of the beam. The gradient follows the distribution of the bending moment which is in terms of shape optimization comprehensible as a design change in the center of the beam has the highest effect on the stresses. The sensitivities in figure 2.5(a) are computed with the forward finite differences step from equation (2.2). As already mentioned in section 2.2.1, the optimal perturbation has to be assessed in advance which is for this example  $1.0 \times 10^{-6}$ . Comparing the gradients in longitudinal direction of the beam, both methods lead to nearly the identical sensitivity field, cf. figure 2.6. Only at the nodes of both ends minor differences can be observed.

This small example demonstrates very well that the implemented adjoint sensitivity analysis of the stress response function from equation (2.30) provides reliable gradient information.



**Figure 2.6:** Gradient field of the stress response function evaluated along the longitudinal direction of the beam. For the purpose of a better comparison, the length of the gradient vectors has been normalized.

## 2.4 Geometrical constraints

Often, in industrial applications the geometrical design space which is available for the evolution of the design is limited. Without the consideration of these constraints, commonly denoted as side constraints, in the definition of the optimization problem, the optimal design does not satisfy them in many cases. A redesign of the optimal shape to comply with the geometric boundary conditions is not trivial and typically leads to a deterioration of the optimized quantities. Therefore, it is essential to integrate the geometric constraints in the optimization algorithm.

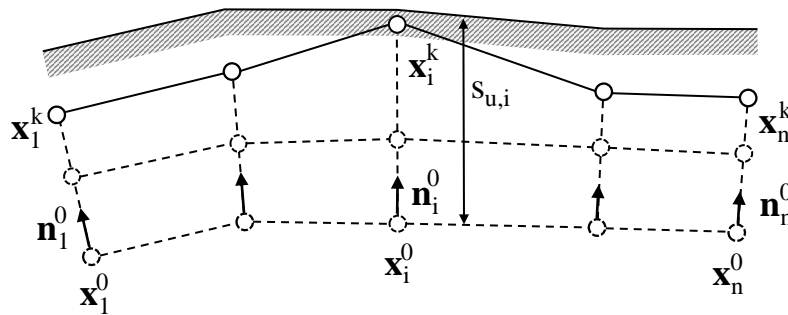
Some of the most significant types of constraints are the thickness of a structure and the bounds of the design variables. For example, for casting structures it is important to limit the maximum wall thickness due to cooling effects. On the other hand, if the members are too thin they cannot be filled with liquid. Both cases can be avoided by a limitation of the wall thickness. Regarding the variable bounds, turbo jet engine components are arranged in a very compact manner due to weight reduction only leading to a limited design space. For the integration of optimized parts in those systems they perfectly have to fit to the existing interfaces.

In many cases the side constraints are simply not included in the optimization algorithm. In case a design variable violates such a constraint during the optimization it is placed directly on the boundary of the constraint. Thus, the resulting design remains feasible which seems very efficient at first sight [Fir10]. Nevertheless, [HG99, Ble90, Roz93] do not recommend this strategy as the abrupt design change leads to oscillations in the convergence history due to the fact that the optimizer tries to violate this constraint in every iteration.

Therefore, in this section, a method is proposed to consider side constraints in a gradient-based optimization algorithm. This approach is applied for the geometric limitation of the design space and the wall thickness constraint.

### 2.4.1 Restricting amount of design change

The evolution of the optimized shape below or above a certain limit can be restrained by the definition of variable bounds  $s_u$  and  $s_l$ , respectively. The approach proposed in this work is discussed in detail on the basis of figure 2.7. Herein, the discretized geometry of the initial design as well as the design at iteration  $k$  is shown and represented by the nodal coordinates of the design variables  $\mathbf{x}^0 = [\mathbf{x}_1^0, \dots, \mathbf{x}_n^0]$  and  $\mathbf{x}^k = [\mathbf{x}_1^k, \dots, \mathbf{x}_n^k]$ . Additionally, every design variable is bounded by an upper constraint  $s_u = [s_{u,1}, \dots, s_{u,n}]$  which is defined as the maximum allowable Euclidean distance between the actual geometry  $\mathbf{x}^k$  and the initial design  $\mathbf{x}^0$  individually for every design variable. Clearly, as the design update takes place in several discrete steps, this bound will not be exactly met. To reduce the risk of a violation of the variable bounds, [Har08] proposed to define them as so-called "smeared" constraints meaning that the variable  $s_i$  is considered to be active if a certain percentage of the permitted maximum displacement  $s_{u,i}$  is reached, e.g.  $\beta = 0.98$ . The "smeared" constraint is highlighted by the grey area in figure 2.7. In the course of the design update from  $k - 1$  to  $k$ , the design variable  $s_i$  attains this area with the consequence that no further displacement in outward direction is allowed.



**Figure 2.7:** Constrained design update with an upper bound  $s_u$  defined for the design variables

For the decision if the geometric constraint is active at node  $s_i$ , the Euclidean distance  $\Delta x_i^k$  between the actual design  $\mathbf{x}_i^k$  and the initial one  $\mathbf{x}_i^0$  has to be computed with

$$\Delta x_i^k = \|\Delta \mathbf{x}_i^k\| = \|\mathbf{x}_i^k - \mathbf{x}_i^0\|. \quad (2.31)$$

Based on this information, the constraint function which reads as

$$\Delta x_i^k - \beta \cdot s_{u,i} \leq 0 \quad (2.32)$$

can be evaluated. As soon as a design variable attains the bound this inequality constraint is violated and the constraint has to be considered as active for this design variable. As the Euclidean distance does not contain any information about the direction of the displacement, the presented procedure is up to this point also valid for lower constraints  $s_l$ .

To determine whether the evolution of the shape at a certain design variable takes place in inward ( $s_l$ ) or outward ( $s_u$ ) direction, the scalar product  $\kappa$  of the shape update vector from equation (2.31) and the normal vector is computed,

$$\kappa_i = \Delta \mathbf{x}_i^k \cdot \mathbf{n}_i^k. \quad (2.33)$$

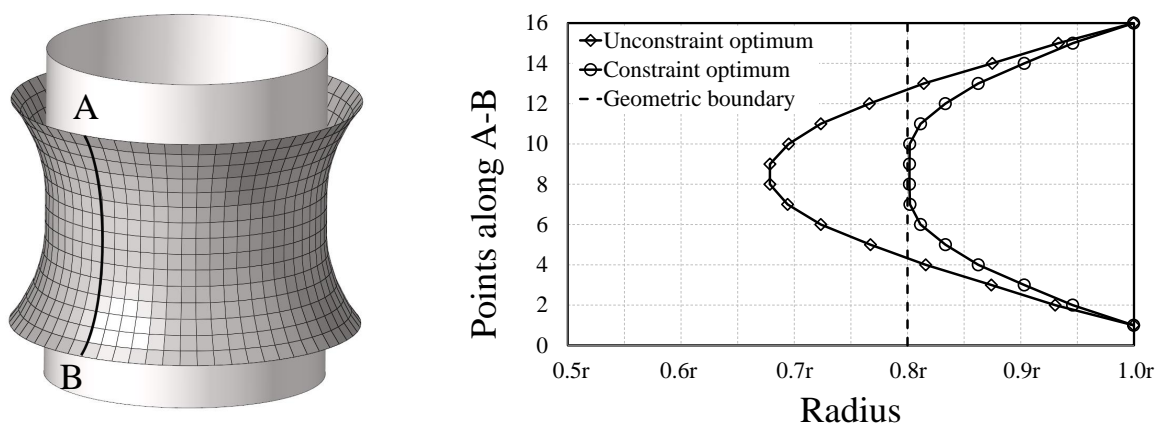
This operation can be interpreted as the projection of the shape update vector to the normal direction. As per definition, the normal vector always points in outward direction for every design variable  $s_i$ , a value of  $\kappa_i > 0$  indicates that the design moves in direction of the upper constraint. Thus, the upper bound  $s_{u,i}$  for the design variable  $s_i$  is activated if equation (2.32) is violated and additionally the vectors  $\Delta \mathbf{x}_i^k$  and  $\mathbf{n}_i^k$  point in the same direction ( $\kappa_i > 0$ ).

As the constraint function (2.32) is linear and it only depends on design variable  $s_i$ , all entries of the gradient vector are zero except for the entry at position  $s_i$  which receives the value one.

$$\begin{bmatrix} s_1 \\ \vdots \\ s_{i-1} \\ s_i \\ s_{i+1} \\ \vdots \\ s_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (2.34)$$

The lower and upper bounds are defined relative to the initial mesh and the Euclidean distance is used to describe the displacement. This means equation (2.32) and the gradient vector in (2.34) also holds for lower constraints  $s_l$ . The only difference is the scalar product  $\kappa$  which has to be negative for lower constraints and positive for upper constraints to activate a constraint. Generally, for a design variable either the upper or the lower constraint can be active but never both at the same time.

In the following, this constraint is tested on the catenoid example introduced in section 2.3.1 which is defined by the radius  $r$  and the height  $h = 1.3r$ . The problem is enhanced by a geometric limitation which bounds the movement of the design variables to a maximum of  $0.2r$ . According to figure 2.8 on the left, this boundary can be interpreted as a cylinder with a radius of  $0.8r$ . It is obvious, that the optimal solution gained in figure 2.3 cannot be reached anymore as it would violate this constraint. The feasible constrained optimum is also shown in figure 2.8 on the left. On the right, the geometries of the unconstrained and constrained optimum are evaluated along the line between point A and B. It clearly can be seen, that in the middle of the constrained catenoid the design variables touch the boundary at the radius  $0.8r$ . This design in fact represents the minimal surface under the defined geometric constraint.

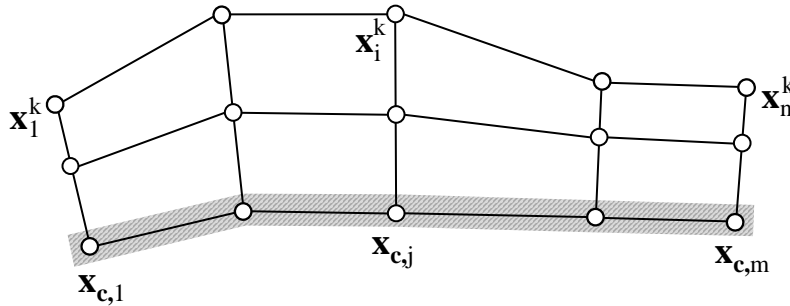


**Figure 2.8:** Catenoid minimum surface under a geometric constraint, constraint optimum (left) and comparisons of unconstrained and constrained geometry (right)

### 2.4.2 Minimum or maximum member size

The second geometric limitation which can be defined in an optimization problem is the minimum or maximum wall thickness. According to the variable bounds introduced in section 2.4.1, the wall thickness is also a linear constraint and can therefore be treated in a comparable manner.

For the derivation of the algorithm let  $\mathbf{x}^k = [\mathbf{x}_1^k, \dots, \mathbf{x}_n^k]$  be the nodal coordinates of the design variables  $\mathbf{s}$  in iteration  $k$  which are modified by the optimizer. To every design variable  $s_i$  a minimum ( $s_{l,i}$ ) and a maximum ( $s_{u,i}$ ) value for the wall thickness can be assigned stored in the vectors  $\mathbf{s}_l$  and  $\mathbf{s}_u$ , respectively. Furthermore, a second node set  $\mathbf{x}_c = [\mathbf{x}_{c,1}, \dots, \mathbf{x}_{c,m}]$  containing the nodal coordinates of  $m$  control nodes is defined. Figure 2.9 visualizes both node sets for an arbitrary mesh topology.



**Figure 2.9:** Constrained design update with an upper bound  $s_u$  for the wall thickness constraint

On that basis the wall thickness is calculated as the distance between the design variables and the control nodes. More precisely, the actual wall thickness  $d_{act,i}$  in iteration  $k$  for the design variable  $s_i$  is defined as the minimum Euclidean distance between  $\mathbf{x}_i^k$  and the control nodes  $\mathbf{x}_c$ ,

$$d_{act,i} = \min_{j=1\dots m} (\|\mathbf{x}_i^k - \mathbf{x}_{c,j}\|). \quad (2.35)$$

The distance  $d_{act,i}$  has to be compared with the minimum or maximum wall thickness  $s_{l,i}$  and  $s_{u,i}$ , respectively. In combination with the safety parameter  $\beta$ ,

introduced in section 2.4.1, the inequality constraint for the maximum allowable wall thickness at node  $s_i$  can be formulated as,

$$d_{act,i} \leq \beta \cdot s_{u,i}, \quad (2.36)$$

and rearranged,

$$d_{act,i} - \beta \cdot s_{u,i} \leq 0. \quad (2.37)$$

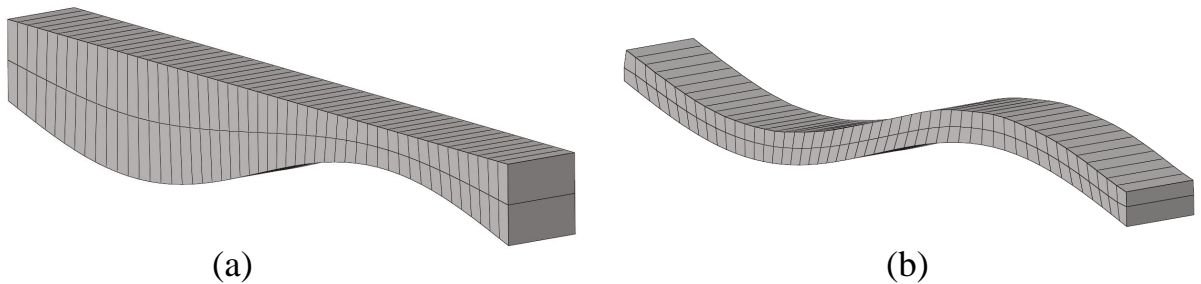
In the same way the constraint for the minimum allowable wall thickness is derived:

$$d_{act,i} \geq \beta \cdot s_{u,i} \quad (2.38)$$

$$\beta \cdot s_{u,i} - d_{act,i} \leq 0 \quad (2.39)$$

In case a design variables exceeds one of these constraints, the assessment of the gradient vector follows exactly the same way as in equation (2.34). Indeed, both constraints from equation (2.36) and (2.38) can be defined for one design variable, but they can never be active at the same time.

Note, this approach is only an approximation of the wall thickness as the normal direction from the node  $s_i$  to the opposite surface would have been exact. However, for highly resolved meshes the distribution of the control nodes  $\mathbf{x}_c$  should be sufficiently dense, that this inaccuracy is negligible.



**Figure 2.10:** Minimum weight of a beam under a minimum wall thickness constraint, initial design (a) and optimal design (b)

To demonstrate the applicability of the proposed wall thickness constraint, a small example is discussed. Figure 2.10(a) shows a beam with a sinusoidal

lower surface and a flat top surface leading to a wall thickness between  $h$  and  $3h$ . The design variables are defined on the top surface acting in vertical direction. The goal is to reduce the mass under the constraint to preserve a minimum wall thickness of at least  $0.8h$  at every position. The corresponding control set is defined on the bottom of the beam. The optimal design which successfully preserves the minimum wall thickness at every design variable is illustrated in figure 2.10(b). As expected, the flat top surface converges to the sinusoidal shape of the lower surface which is the constrained optimal design.

## 2.5 Sensitivity weighting

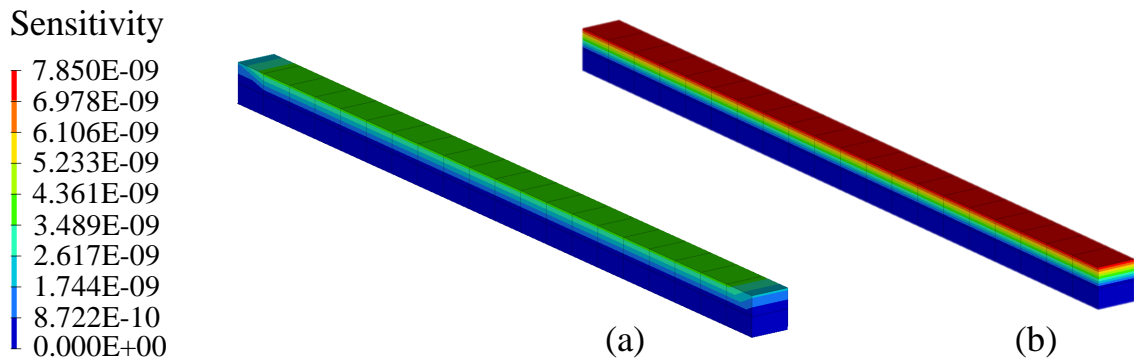
A general problem in discrete sensitivity analysis is the fact that the shape sensitivities strongly depend on the discretization. The sensitivity value at each node inherently reflects the influences of the surrounding mesh and the element formulation [Fir10, KSWB14]. In case of gradient-based optimization methods this disturbance in the gradient information should be controlled carefully. Therefore, in this section, these two major sources of inaccuracies are investigated and practical remedies are presented.

### 2.5.1 Dependency of the discretization

The influence of the discretization is deduced based on the example of figure 2.4. The objective is to minimize the mass of the beam according to equation (2.21) evaluated for the whole structure. The mechanical model is regularly discretized with two rows of linear 8-node hexahedral elements with full integration (Fig. 2.4(b)). The design variables, defined at the top of the structure, are perturbed in vertical direction.

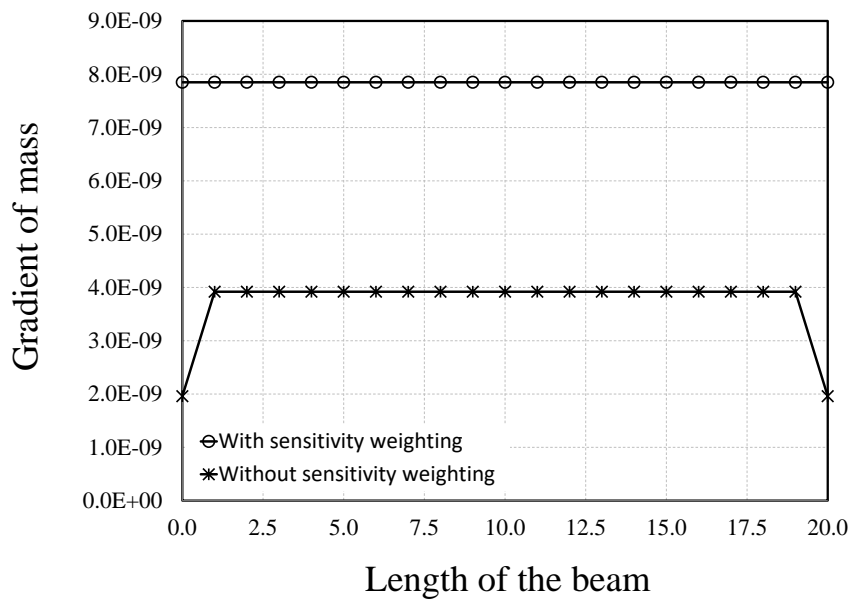
Regarding the gradient field of the mass response function, it is expected to be constant, since a movement of any material point should have the identical impact on the weight of the structure. Keeping this in mind, at first the gradients without any additional weighting step are calculated and illustrated in figure 2.11(a). The sensitivity field does not show the expected constant distribution since the values





**Figure 2.11:** Gradient field of the mass response function without sensitivity weighting (a) and with sensitivity weighting (b)

at the corner nodes are by a factor of two lower than the inner ones, cf. figure 2.12. This inaccuracy is introduced by the discrete character of the sensitivities. To counteract this effect, weighting is proposed as additional post-processing step leading to the desired constant behavior as figure 2.11(b) clearly shows. The



**Figure 2.12:** Gradient field of the mass response function evaluated along the longitudinal direction of the beam

evaluation of the gradients along the longitudinal direction of the beams in figure 2.12 reveals the main differences. The nodal sensitivities in the version with weighting represent exactly the defined density of  $7.85 \times 10^{-9}$  being the correct value. The nodal values in the version without the weighting step still have to be multiplied with a certain factor to achieve the desired result of version (b). In this

example the scaling factor turns out to be 4 for the corner nodes ( $3.925 \times 10^{-9}$ ) and 2 for the interior ones ( $1.963 \times 10^{-9}$ ). Applying this weighting step, the sensitivity field is considered improved in the sense that it now reflects the characteristics of the underlying optimization problem independent of the chosen discretization.

To understand the derivation of these weighting factors in detail, let us consider an arbitrary objective function  $J$  which depends on the position of the discrete control point parameters  $s_l$  through the surface geometry  $\mathbf{S}$  and the "objective density"  $\phi$ :

$$J(\mathbf{s}_l) = \int_A \phi(\mathbf{S}(\mathbf{s}_l)) dA = \int_A \phi \left( \sum_{l=1}^{n \times m} R_l(\xi, \eta) \mathbf{s}_l \right) dA \quad (2.40)$$

Herein,  $R(\xi, \eta)$  is the two-dimensional shape function which discretizes the surface geometry  $\mathbf{S}$  and  $n \times m$  is the number of the control points distributed on the area  $A$  on the surface element. According to the chain rule of differentiation, the sensitivities with respect to the control points  $\mathbf{s}_l$  can be expressed as

$$\nabla_s J = \int_A \frac{\partial \phi}{\partial \mathbf{S}} \frac{\partial \mathbf{S}}{\partial \mathbf{s}_l} dA = \int_A \frac{\partial \phi}{\partial \mathbf{S}} R_l(\xi, \eta) dA. \quad (2.41)$$

From equation (2.41) it is clear that the geometric influence of the mesh on the sensitivity is defined by the integral of the corresponding shape functions. Thus, the discrete sensitivities reflect the density of the discretized mesh [KSWB14]. The sensitivity values  $\nabla_s J$  are of integral nature whereas the design variables  $\mathbf{s}$  itself are the supporting values of a field. For the sake of consistency the sensitivity values also have to be of "field nature" which is achieved by computing the geometric influence coefficient  $b_l$  for every design variable and scaling the sensitivity values with the inverse of the geometric influence matrix  $\mathbf{B}$ :

$$b_l = \int_A R_l(\xi, \eta) dA \quad (2.42)$$

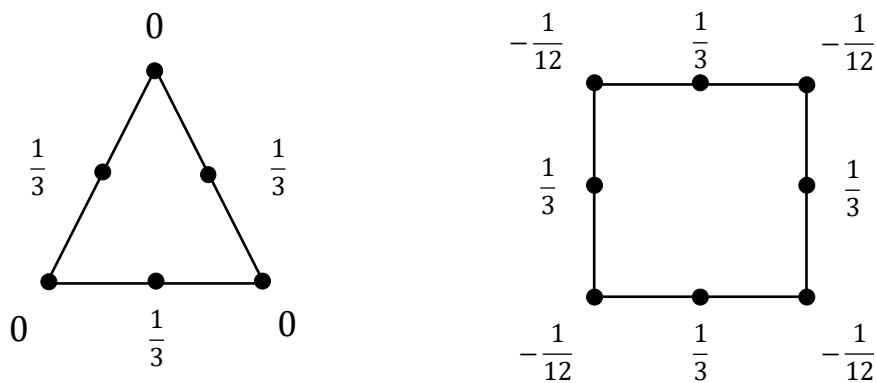
$$\mathbf{B} = \text{diag}(b_l) \quad (2.43)$$

$$\nabla_s J_w = \mathbf{B}^{-1} \nabla_s J \quad (2.44)$$

It is important to note, taking the weighted sensitivities  $\nabla_s J_w$  does not alter the optimization problem since only the search direction is modified. Consequently, for non-convex problems both weighted and unweighted sensitivities lead to the same optimum solution. But in case of non-convex problems different local optima might be obtained.

### 2.5.2 Influence of the element formulation

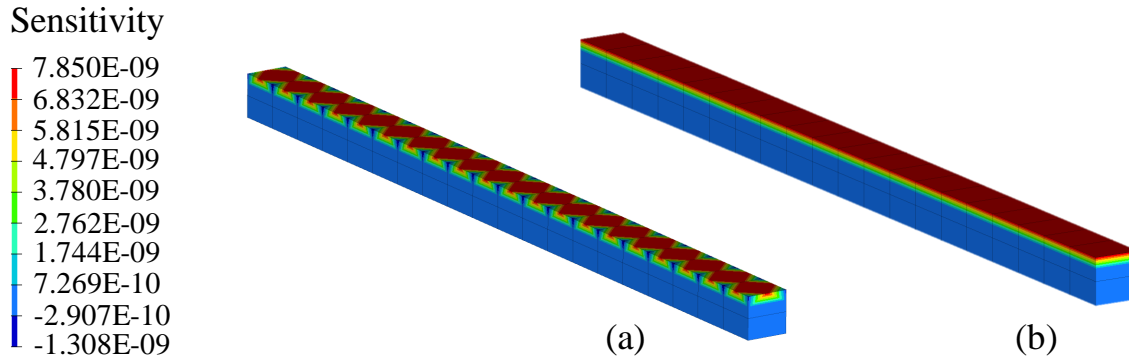
Apart from the discretization, the element formulation itself influences the gradient field. In structural simulations quadratic elements are typically used. Therefore, the example of figure 2.4 has been assessed again with 20-node hexahedral elements. The computation of the geometric influence matrix  $\mathbf{B}$  from equation (2.43) reveals the drawback of the higher order elements. As figure



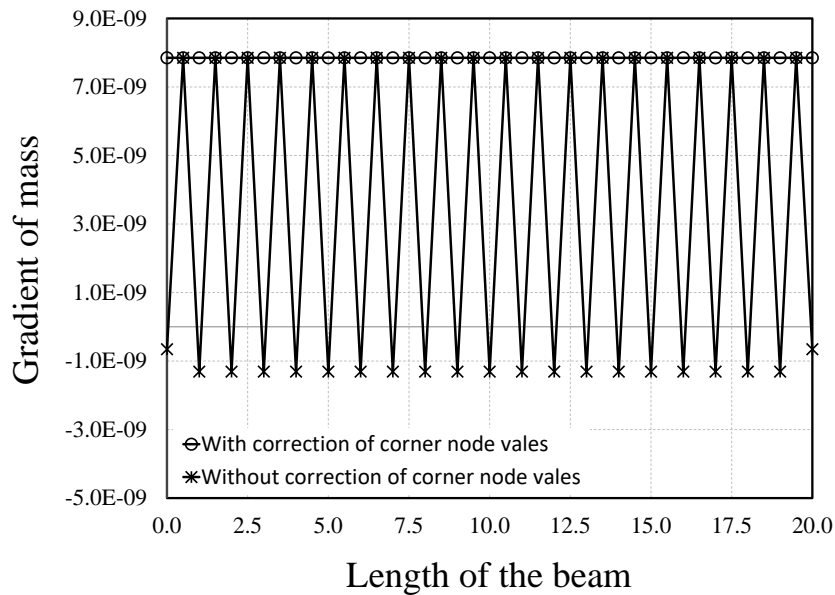
**Figure 2.13:** Geometric influence coefficients  $b_l$  for unit tetrahedral and hexahedral elements

2.13 shows, its diagonal entries  $b_l$  are negative for hexahedral elements or even zero for tetrahedral ones. Applying the  $\mathbf{B}$ -matrix on the quadratic hexahedral elements leads to a sensitivity field which is alternating between the corner nodes and the mid nodes of the element from negative to positive values. The figures 2.14(a) and 2.15 clearly show the described behavior. The negative values at the corner nodes are physically wrong as they suggest a weight reduction in outward direction whereas the values at the mid nodes are correct showing the desired

positive sensitivity value of  $7.85 \times 10^{-9}$  of the mass response functions.



**Figure 2.14:** Gradient field of the mass response function for quadratic hexahedral elements without (a) and with (b) correction of the values at the corner nodes



**Figure 2.15:** Gradient field of the mass response function evaluated along the longitudinal direction of the beam

As a practical remedy the sensitivities at the corner nodes are not computed at all in the sensitivity analysis but are extrapolated after the weighting step in equation (2.44) from the neighboring mid-node values:

$$\nabla_s J_{w,cor} = \frac{1}{n} \sum_{i=1}^n \nabla_s J_{w,mid,i} \quad (2.45)$$

Herein,  $n$  is the number of mid nodes next to the considered corner node derived from the topology of the mesh. With this extrapolation the expected constant sensitivity field can be obtained, see figures 2.14(b) and 2.15.

Neglecting the sensitivities at the corner nodes reduces the number of design variables at a first sight. However, these short-waved solutions where the mid node moves independent from the corner node are not desired anyway and are eliminated by the out-of-plane regularization (section 2.6.3). Thus, it can be assumed that the interpolation step for quadratic elements does not restrict the design space remarkably.

## 2.6 Regularization of shape optimization problems

In this section the regularization methods in node-based shape optimization are discussed. The section is organized as follows: Section 2.6.1 introduces the main problems in node-based optimization and points out the necessity of regularization. The various out-of-surface regularization methods are discussed in section 2.6.2 and the proposed approach in the Vertex Morphing method is presented in 2.6.3. Section 2.6.4 introduces the different regularization methods for retaining the quality of the volume mesh during the optimization. At the end of this section (section 2.6.5), an extended Traction method which is utilized for update of the three-dimensional mesh in the Vertex Morphing method is presented.

### 2.6.1 Motivation

Shape optimization problems of linear elastic bodies in equilibrium can be generalized as optimization problems of domains in which elliptic boundary value problems are defined [AKSK97]. It is well known that this kind of domain optimization problems do not have sufficient regularity [HG99, BWDC05]. According to [Fir10] the methods to overcome this ill-posedness can be separated in techniques for the out-of-surface regularization and for the regularization of the volume mesh, commonly denoted as "in-plane regularization" in the case of

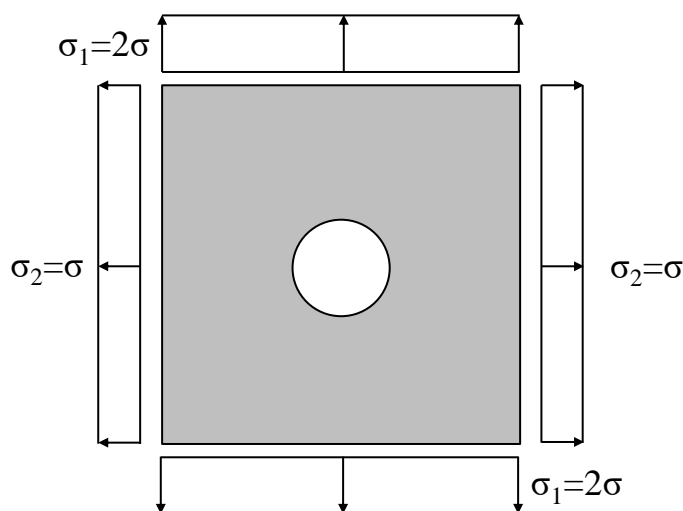
two-dimensional shell structures.

Regarding the out-of-surface regularization, independent from the type of differentiation (analytical, semi-analytical or global finite differences) the gradients of the response functions are not smooth. In this context the term "smooth" is related to a geometrical curvature measure as in gradient-based shape optimization problems the smoothness of the gradients is directly linked to the smoothness of the optimized geometry. This non-smoothness mainly arises from the fact that the computed gradients are less regular than the parametrization [Sta15, JMP98, MP09]. In this context, Haslinger et al. [HN88] worked out that an optimal shape for elliptic boundary value problems can only be achieved if the constraint of Lipschitz continuity is fulfilled as a requirement for the smoothness of the shape. However, this has not been discussed thoroughly and there is still a lack of theory. To overcome this irregularity several approaches are existing: Imam [Ima82] suggests to limit the number of design variables by isoparametric representations of the surfaces which corresponds with the idea of Braibant et al. [BF84] to use B-splines for the definition of the geometry. Among others Bletzinger [Ble13], Schmidt [Sch10] and Le et al. [LBT11] ensure a well-posed shape optimization problem by filtering techniques that imposes a minimum length scale. Independent of the chosen method, there is in fact a need for an out-of-surface regularization.

During the optimization the finite element mesh typically undergoes big design changes and therefore can easily get distorted. The volume mesh regularization is applied to retain a sufficient mesh quality and particularly to preserve reliable gradient information in every optimization iteration. Hence, the goal of these methods is to change the discretization of the geometry and not the geometry itself. In case of node-based shape optimization the nodal positions of the mesh are adjusted.

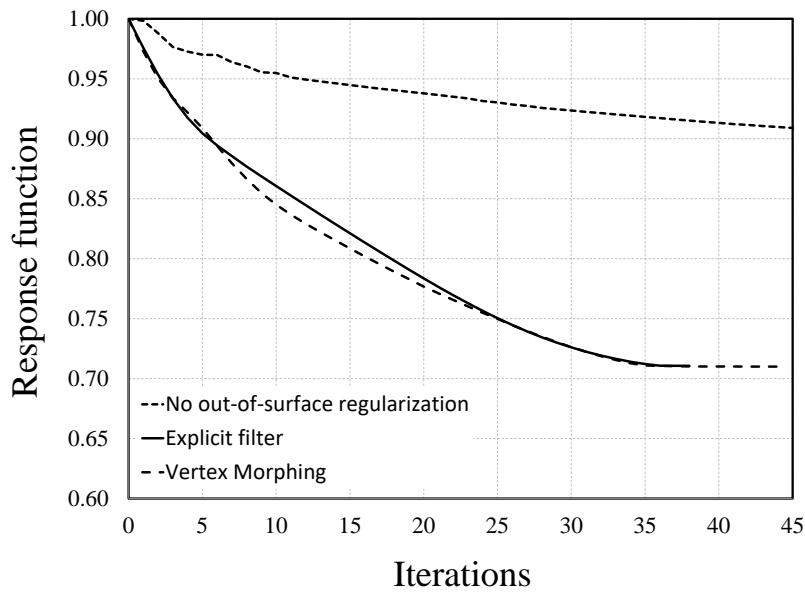
The Vertex Morphing method presented in this work contains both the out-of-surface regularization and the regularization of the volume mesh. In order to evaluate the proposed methods with respect to the smoothness of the sensitivity

field and the quality of the grid during the optimization the two-dimensional quadratic plate with a circular hole of figure 2.16 is considered. The objective is the reduction of the notch stresses with equation (2.25) as response function. The nodes on the edge of the hole are defined as design variables and additionally evaluated for the response function. Furthermore, as a constraint the mass of the plate must not exceed the initial mass. As one can expect, the optimizer will update the circular hole to an elliptical one with a ratio of the axis of 2/1 which is identical to the ratio of the external loading. Under the defined constraint this is the optimal design as the stress concentration is minimized and homogenized around the edge of the hole.



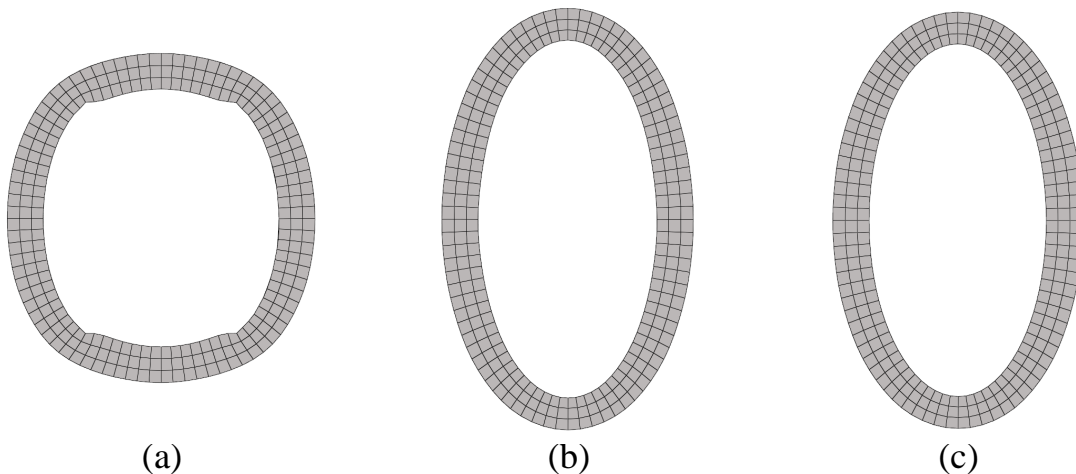
**Figure 2.16:** 2D quadratic plate with circular hole under tension stresses

Regarding the out-of-surface regularization three different cases are studied for the sake of comparison. In the first one, no regularization is applied at all. In the second case an explicit filter is applied to the computed gradient field as described in section 2.6.2 and in the third one the regularization of the Vertex Morphing method is applied (section 2.6.3). Figure 2.17 compares the reduction of the objective function during the optimization as an indication for the quality of the gradients received from the out-of-surface regularization step. The function values are normalized, therefore a value of 1 relates to the initial design. In the first case where no additional regularization of the gradient field takes place, the optimization did not even finish after 100 iterations with an improvement of the objective function by less than 10%. The other two cases converged to the optimal design after 38 and 44 iterations with an improvement of 29% in both



**Figure 2.17:** Comparison of out-of-surface regularization methods with respect to the improvement of the normalized response function

cases. According to figure 2.18 the optimal design of case one did not converge to the analytical solution but sticks at a the circular shape forming several kinks. The final shapes of case two and three develop the expected elliptical shape with the ration of 2/1. Though an additional effort has to be given, the out-of-surface regularization is mandatory for node-based parametrizations.

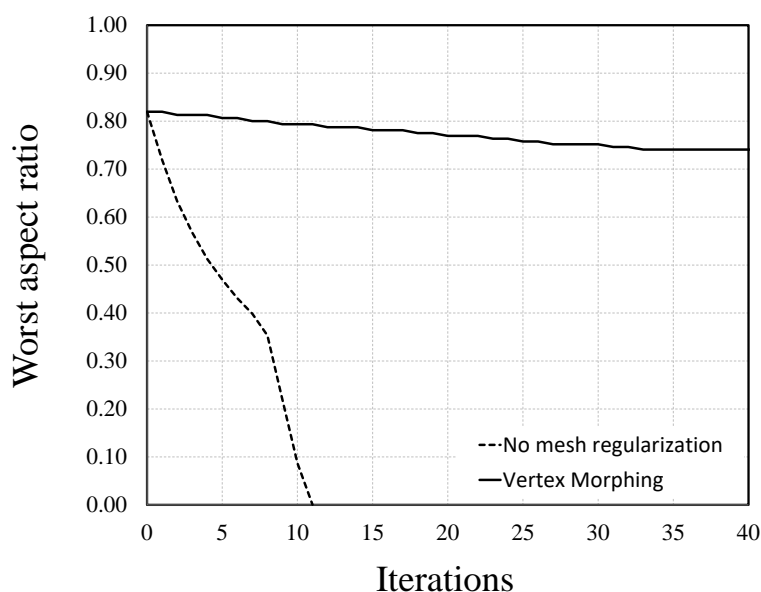


**Figure 2.18:** Comparison of optimal shapes of the circular hole for an optimization without any out-of-surface regularization (a), with an explicit filter (b) and the Vertex Morphing method (c)

At a first sight, the optimal design seems to be trivial but the reduction of



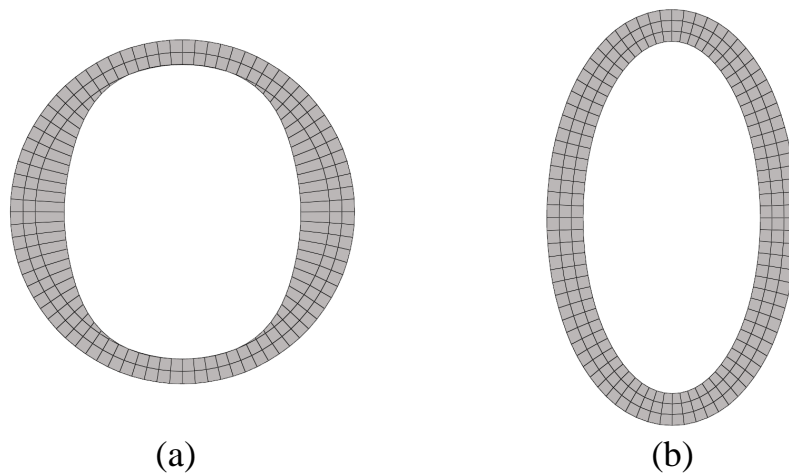
curvature and the update of the interior nodes is a very challenging task in shape optimization since the elements easily get distorted if no volume mesh regularization is applied. Therefore, this example is also the basis to demonstrate the importance of controlling the volume mesh. In the first case, the improvement of the mesh quality is neglected only the smoothing of the sensitivity field with an explicit filter applied. In the second case Vertex morphing which is based on the traction method [AW96] is applied, cf. section 2.6.5.



**Figure 2.19:** Comparison of volume mesh regularization methods with respect to the change of the aspect ratio

Figure 2.19 compares the mesh quality during the optimization for the different cases. The figure shows the quality of the elements located around the circular hole as these are the ones where the sensitivities are computed and the shape change mostly takes place. For a reliable gradient information the decrease of the element quality should be as less as possible. As a quality measure the aspect ratio of the worst element is evaluated. It is defined as the ratio between the shortest and the largest element edge. A perfectly shaped element has the value of 1 whereas a ratio of 0 indicates that the element already collapsed since at least one element edge approached zero. In the case where no additional volume mesh regularization is applied the optimization failed at iteration 9. As can be seen in figure 2.20(a), the elements at the top and at the bottom of the ellipse collapsed. The Vertex Morphing method with volume mesh regularization

continued until the optimum design was reached, see figure 2.20(b). The Vertex Morphing method delivers a higher quality of the mesh and thus allows for more optimization steps with the drawback of an increased computational effort. However, if large design changes are expected a mesh quality preservation step is mandatory in the optimization process.



**Figure 2.20:** Comparison of final shapes of the circular hole for an optimization without any volume mesh regularization (a) and with regularization (b)

In what follows, an overview of the common regularization methods is given and in particular the methods implemented in the Vertex Morphing method are presented. For the out-of-surface regularization of the sensitivity field a consistent formulation is deduced where the smoothing is no longer considered as a "post-processing" step. The method for the volume mesh regularization is based on the ideas of the Traction method. To further reduce the mesh deterioration, two extensions are included: in the pseudo-elastic problem the invariable areas are only constrained in normal to the surface direction, i.e. the direction responsible for the design change. Tangentially, the nodes can freely vary. Furthermore, the Young's modulus is defined as a function of the distance from the design variables. This leads to a variable stiffness in the component which distributes the displacements more evenly in the finite element mesh and further reduces the mesh dependencies.

## 2.6.2 Overview of out-of-surface regularization methods

As mentioned in the section above, almost all gradient fields in node-based shape optimization problems are not smooth. For this reason a projection, smoothing or filtering step is required [MP09, JV00, LBT11]. Generally, the methods used for the regularization of the sensitivity field can be explicit or implicit. The explicit methods are applied directly on the noisy field while the implicit ones operate on the smooth (unknown) field [Sta15].

Explicit filters are based on convolution integrals [Yos95] where the raw sensitivity field  $G$  is convoluted with a kernel  $K$  to obtain the smooth field  $\bar{G}$ :

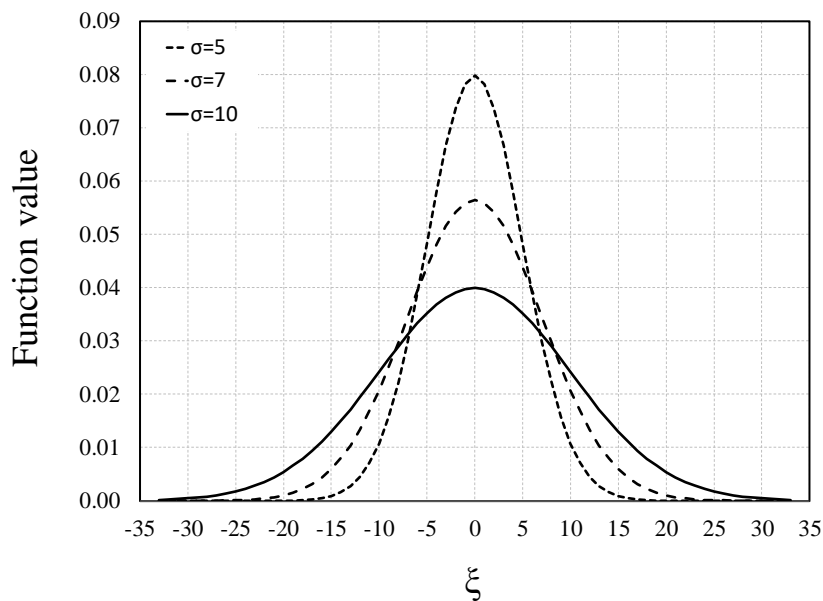
$$\bar{G}(\xi_0) = \mathcal{A}G(\xi) = \int_{\Gamma} K(\xi)G(\xi_0 - \xi)d\xi \quad (2.46)$$

Herein,  $\mathcal{A}$  and  $\xi$  represent the filtering operator and the local curved coordinate, respectively. Mind, the description is one-dimensional but an extension to two-dimensional surfaces with a local system of orthogonal surface coordinates is straight-forward. Motivated by scale-space theory [Lin94], Gaussian kernels are commonly used in shape optimization problems due to their scale-space properties like linearity, shift invariance, non-enhancement of local extrema and invariance of scale and rotation [SR11]:

$$K(\xi) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{|\xi|^2}{2\sigma^2}} \quad (2.47)$$

In this equation  $|\xi|$  is the Euclidean distance to the center of the filter. The standard deviation  $\sigma$  is denoted as filter radius and assumed to be constant during an optimization. The higher the value of  $\sigma$ , the wider the influence of the filter and thus the gradient field becomes smoother, cf. figure 2.21.

Regarding the class of implicit filters, a well established and commonly used filter in shape optimization of structural and aerodynamic problems is the Sobolev-gradient smoothing [JV00, MP09, SIGS08, JJ07, Sch10, Moh97]. In



**Figure 2.21:** Three different realizations of Gaussian kernels with  $\sigma = 5$ ,  $\sigma = 7$  and  $\sigma = 10$ .

this method, the Laplace-Beltrami operator

$$I - \epsilon \frac{\partial}{\partial x} \frac{\partial}{\partial x} \quad (2.48)$$

is applied to compute the smoothed gradient field  $\bar{G}$  as the solution of the following elliptical equation

$$\bar{G} - \epsilon \frac{\partial}{\partial x} \frac{\partial \bar{G}}{\partial x} = G. \quad (2.49)$$

The parameter  $\epsilon$  is an arbitrary positive scalar value penalizing high curvatures detected by the curvature operator  $\partial^2 / \partial x^2$ . It can be noted that equation (2.48) can also be applied directly to the design update  $\delta s$ .

In the following, the relation between the explicit and implicit filters are shown for the one-dimensional case. According to [SR11], the raw sensitivity field  $G$  convoluted with a Gaussian kernel with the filter  $\sigma^2 = \gamma t$  can be considered as a fundamental solution of the unsteady diffusion equation

$$\frac{\partial \bar{G}}{\partial t} = \frac{\gamma}{2} \cdot \frac{\partial^2 \bar{G}}{\partial x^2}. \quad (2.50)$$

The term  $\gamma/2$  is the diffusion coefficient indicating how fast the diffusion of  $\bar{G}$  is over the pseudo-time  $t$ . For equation (2.50) an implicit first-order approximation is stated in [SR11] and [JJ07] which yields

$$\bar{G} - \frac{\gamma t}{2} \frac{\partial^2 \bar{G}}{\partial x^2} = \bar{G}(0). \quad (2.51)$$

Comparing equation (2.49) and (2.51) leads to the relation

$$\sigma^2 = 2\epsilon, \quad (2.52)$$

so that the smoothing intensity  $\epsilon$  can be interpreted as half the variance  $\sigma^2$  from the Gaussian kernel. Hence, an explicit filtering with an Gaussian kernel is first-order equivalent to the implicit Sobolev-gradient smoothing.

Generally, the optimal choice of  $\epsilon$  and  $\sigma$  is case dependent and can be considered as a Newton's step in the steepest descent method of equation (2.71) [SIGS08]. But the determination of the optimal values for both parameters requires additional computational effort. Therefore, in practice they are often considered as additional design variables guiding the optimal design to the desired curvature.

The explicit and implicit filter methods have in common that they are tuned by a single scalar value, the "filtering coefficient" which indicates the "filtering intensity". Furthermore, both smooth the design by establishing a distance-based relation between the different nodal values. Practically, the choice of the filter method is not as decisive as the choice of the filtering coefficient.

### 2.6.3 Consistent out-of-surface regularization

The filter methods presented in the previous section smooth the sensitivity field in an additional step decoupled from the optimization problem. In this section, the filtering in the Vertex Morphing method [Ble13, HSB14, Hoj14] is introduced where the regularization of the sensitivity field is no longer considered as a "post-processing" step but is consistently incorporated in the optimization problem. Therefore, beside the geometry field  $x$  an additional control field  $s$  is

introduced on which the mathematical optimization problem is defined. Both fields are related by a mapping operator  $\mathcal{A}$ . Furthermore a material coordinate  $\xi$  (one-dimensional case) is defined on  $\Gamma$ . Both the geometry field  $x(\xi)$  and the control field  $s(\xi)$  are a function of this material coordinate.

The geometry  $x$  is obtained from the control field  $s$  by the following operation,

$$x(\xi_0) = \mathcal{A}s = \int_{\Gamma} A(\xi_0, \xi) s(\xi) d\xi. \quad (2.53)$$

In this equation the geometry  $x$  is obtained by a convolution of the control field  $s$  with a kernel  $A$  which can be the Gaussian kernel  $K$  from equation (2.47) or any other self-adjoint function. Typically, in the beginning of an optimization an initial geometry  $x_0$  is given and the distribution of the related design control field  $s_0$  is unknown. However, the absolute values of the design control field don't have to be known as one is only interested in its change  $\delta s$  to determine the change of the geometry  $\delta x$ . According to equation (2.53) the variation of the geometry is formulated as

$$\delta x(\xi_0) = \int_{\Gamma} A(\xi_0, \xi) \delta s(\xi) d\xi. \quad (2.54)$$

Based on equation (2.53), Bletzinger [Ble13] concluded that the derivative of  $x(\xi_0)$  with respect to  $s(\xi)$  equals  $A(\xi_0, \xi)$ ,

$$\frac{dx(\xi_0)}{ds(\xi)} = A(\xi_0, \xi) \quad (2.55)$$

Substituting equation (2.55) in (2.53) the change of the geometry can also be expressed as

$$\delta x(\xi_0) = \int_{\Gamma} \frac{dx(\xi_0)}{ds(\xi)} \delta s(\xi) d\xi. \quad (2.56)$$

In gradient-based shape optimization one is interested in the change of the objective function  $J(x, u)$  with respect to the design  $s$ . Applying the chain rule of differentiation the following expression can be derived:

$$\frac{dJ}{ds} = \frac{\partial J}{\partial s} + \frac{\partial J}{\partial x} \frac{\partial x}{\partial s}. \quad (2.57)$$

The term  $\partial J/\partial s$  is zero since a change of the control field does not alter the objective function. Then, equation (2.55) can be substituted in (2.57) which finally yields

$$\frac{dJ}{ds}(\xi_0) = \int_{\Gamma} \frac{\partial J}{\partial x} A(\xi, \xi_0) d\xi. \quad (2.58)$$

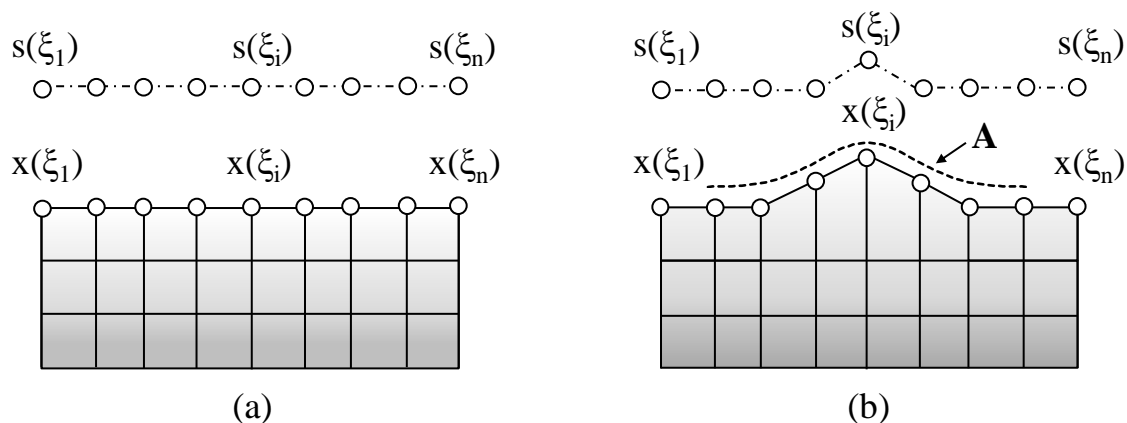
In this equation only the partial derivative  $\partial J/\partial x$  remains. This means, the change of the objective function  $J$  with respect to the control field  $s$  is purely based on the variation of the geometry  $x$ . This term can efficiently be computed by means of the adjoint sensitivity analysis introduced in section 2.2. According to [Ble13] this procedure is defined as "sensitivity filtering" as higher oscillating geometrical modes are suppressed.

The main difference between Vertex Morphing and the aforementioned filter methods is the consistently in the optimization problem included filtering step by the chain rule of differentiation. That means, the filter appears twice in an optimization step: once for the update of the geometry in equation (2.54) and a second time for the computation of the sensitivities in equation (2.58).

Typically, in numerical applications the geometry is discretized and consequently the control field as well, cf. figure 2.22(a). The strategy here is to discretize the control field  $\mathbf{s} = [s_1, \dots, s_n]$  with as many nodes as the geometry space  $\mathbf{x} = [x_1, \dots, x_n]$ . Similar to the  $\mathcal{A}$  operator from equation (2.46), a matrix  $\mathbf{A}$  can be defined which linearly maps  $\mathbf{s}$  onto  $\mathbf{x}$ :

$$\mathbf{x} = \mathbf{A}\mathbf{s} \quad (2.59)$$

Figure 2.22(b) shows the change of the geometry  $\mathbf{x}$  due to a change of the control field at the position  $s_i$  linked by the  $\mathbf{A}$  operator. If  $\mathbf{x}$  and  $\mathbf{s}$  have the identical number of grid points and these are distributed equally,  $\mathbf{A}$  is symmetric and  $\mathbf{A}^T = \mathbf{A}$ . But this does not necessarily have to be the case. Thinking about CAD parametrization, the number of control points is smaller than the number of geometry parameters. Nevertheless, the role of the control field remains identical between the FE and CAD parametrization.



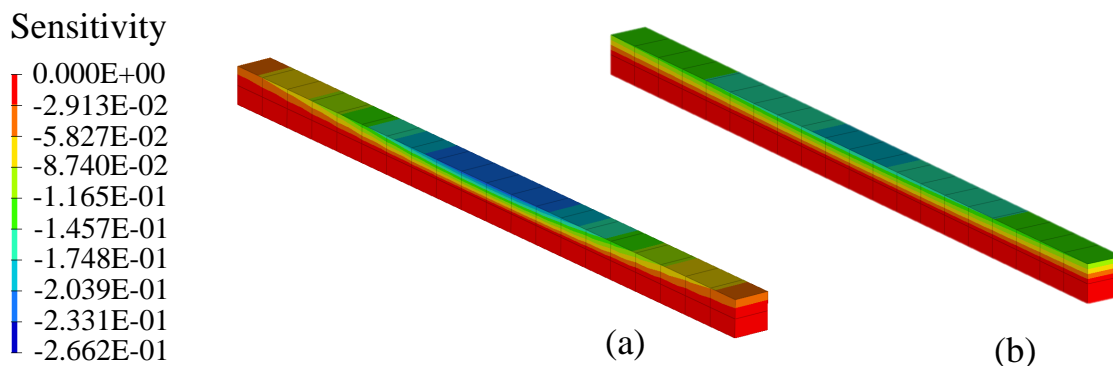
**Figure 2.22:** Discrete geometry  $\mathbf{x}$  and control field  $\mathbf{s}$  (a) and mapping between  $s(\xi_i)$  and  $x(\xi_i)$  with the filter matrix  $\mathbf{A}$  (b)

According to equations (2.54) and (2.57) the formulas for the variation of the geometry  $\delta \mathbf{x}$  and the computation of the sensitivities  $\nabla_{\mathbf{s}} J$  can also be written in a discrete form,

$$\delta \mathbf{x} = \mathbf{A} \delta \mathbf{s}, \quad (2.60)$$

$$\nabla_{\mathbf{s}} J = \mathbf{A}^T \nabla_{\mathbf{x}} J. \quad (2.61)$$

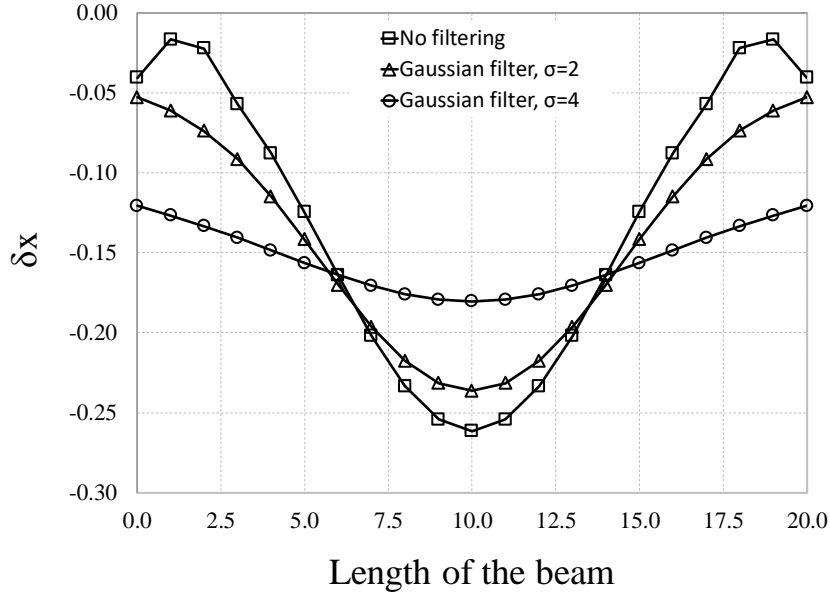
Again, in the above equations it becomes clear, that the filter has to be applied twice in an optimization step. Once to generate the geometry  $\mathbf{x}$  or its variation  $\delta \mathbf{x}$  from the control field  $\mathbf{s}$  or  $\delta \mathbf{s}$ , respectively and once to filter the nodal sensitivity  $\nabla_{\mathbf{x}} J$  back to the control field  $\nabla_{\mathbf{s}} J$ . In most of the cases  $\mathbf{A}$  is symmetric thus the backward filtering in equation (2.60) is equal to the forward filter operation.



**Figure 2.23:** Variation of the geometry  $\delta x$  for the model from figure 2.4 based on a Gaussian filter with a filter radius of  $\sigma = 2$  (a) and a filter radius of  $\sigma = 4$  (b)



Figure 2.23 shows the design update of the geometry  $\delta x$  for the mechanical model from figure 2.4 for two different filter radii computed with the Vertex Morphing method. The filter function is based on the Gaussian kernel with a filter radius of  $\sigma = 2$  and  $\sigma = 4$  applied on the gradients of the stress response function of equation (2.30). Figure 2.24 compares these results with the one where no filter is applied (figure 2.5(b)) in longitudinal direction of the beam. It can be seen, that the local small scale oscillations at the corners and the maximum in the middle are smoothed. Furthermore, the higher the filter radius, the lower the curvature of the design update.



**Figure 2.24:** Comparison of the variation of the geometry  $\delta x$  with different filter radii for the model from figure 2.4

To study the effect of the formulation of the geometry in terms of  $\mathbf{s}$  instead of  $\mathbf{x}$  on the solution of the optimization problem, the second order Taylor series expansion of the objective function is considered,

$$\tilde{J} = J + (\nabla_{\mathbf{s}} J)^T \delta \mathbf{s} + \frac{1}{2} \delta \mathbf{s}^T \mathbf{H}_{\mathbf{s}} \delta \mathbf{s}. \quad (2.62)$$

Herein,  $\mathbf{H}_{\mathbf{s}}$  denotes the Hessian matrix with respect to the control parameters  $\mathbf{s}$ .

The derivative of  $\tilde{J}$  with respect to  $\delta\mathbf{s}$  can be computed as

$$\nabla\tilde{J} = (\nabla_{\mathbf{s}}J)^T + \mathbf{H}_{\mathbf{s}}\delta\mathbf{s}, \quad (2.63)$$

and the stationary condition  $\nabla\tilde{J}_{\mathbf{s}} = 0$  of this equation can be solved for  $\delta\mathbf{s}$ ,

$$\delta\mathbf{s} = -\mathbf{H}_{\mathbf{s}}^{-1}\nabla_{\mathbf{s}}J. \quad (2.64)$$

Now, the problem is be reformulated in terms of  $\mathbf{x}$  with the following two relations:

$$\nabla_{\mathbf{s}}J = \mathbf{A}^T\nabla_{\mathbf{x}}J, \quad (2.65)$$

$$\mathbf{H}_{\mathbf{s}} = \mathbf{A}^T\mathbf{H}_{\mathbf{x}}\mathbf{A}. \quad (2.66)$$

Substituting the equations (2.65) and (2.66) in (2.62) leads to the design update  $\delta\mathbf{s}$  in terms of the control field,

$$\delta\mathbf{s} = -(\mathbf{A}^T\mathbf{H}_{\mathbf{x}}\mathbf{A})^{-1}\mathbf{A}^T\nabla_{\mathbf{x}}J. \quad (2.67)$$

Finally the variation of the geometry  $\delta\mathbf{x}$  is obtained by replacing  $\delta\mathbf{s} = \mathbf{A}^{-1}\delta\mathbf{x}$  in equation (2.68),

$$\delta\mathbf{x} = \mathbf{A}(-\mathbf{A}^{-1}\mathbf{H}_{\mathbf{x}}^{-1}\mathbf{A}^{-T})\mathbf{A}^T\nabla_{\mathbf{x}}J = -\mathbf{H}_{\mathbf{x}}^{-1}\nabla_{\mathbf{x}}J. \quad (2.68)$$

In the above equation the filter matrix  $\mathbf{A}$  finally cancels out and independent of the parametrization the optimization problem remains unchanged. This implies that for convex problems the optimal design is not influenced by the choice of  $\mathbf{A}$  but only the geometry is The reparameterized by a variable transformation.

As a matter of fact in most of the engineering applications the optimization problem is non-convex and the filter influences the optimal design. The goal is to find an optimal solution which satisfies both the requirements defined in the optimization problem and the manufacturing constraints with respect to the smoothness. The design of the different local minimum is characterized by specific curvature meaning that there is always an operator  $\mathbf{A}$  defined through the filter radius which corresponds to such a design. Consequently, the filter radius

can be considered as an additional design handle to steer the optimal design and to explore the whole design space.

### 2.6.4 Overview of mesh quality control methods

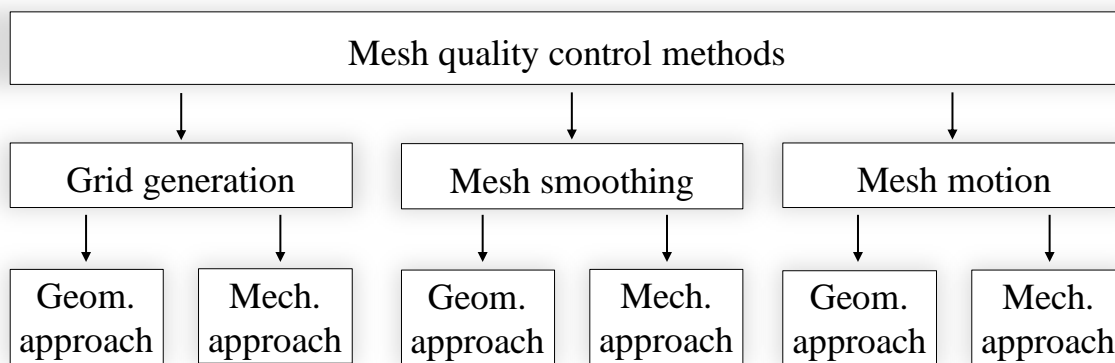
Generally, in node-based shape optimization problems the geometry is described by the Finite Element mesh and no other link to any CAD description is preserved. In this discretization the nodes located at the surface of the mesh are considered as the design variables of the optimization problem since only these nodes are responsible for a design change. A movement of the internal nodes does not alter shape but only the discretization. More precisely, the design change of the surface nodes can be decomposed in a normal and a tangential direction with respect to the surface,

$$\delta \mathbf{x}_l = \delta x_{l,n} \cdot \mathbf{n} + \delta x_{l,t} \cdot \mathbf{t} = s_l \cdot \mathbf{n} + r_l \cdot \mathbf{t}, \quad l = 1, \dots, n_s. \quad (2.69)$$

Here,  $n_s$  is the number of surface nodes,  $\mathbf{n}$  and  $\mathbf{t}$  are the unit update vectors normal and tangential to the surface at node  $l$ . Due to the decomposition it becomes clear,  $s_l$  is the shape relevant component which is responsible for the design change. Consequently this component is defined as design variable and updated by the optimizer. The tangential component  $r_l$  only changes the discretization and is called the mesh relevant component [Sta15]. The update of this component together with the position of the internal nodes is addressed in the volume mesh regularization methods. As demonstrated in the example in figure 2.20 the mesh can easily become distorted if the focus is only on the update of the out-of-surface direction. In these cases the optimization often aborts before the optimum is reached. Hence, a method for the mesh quality control is strongly recommended.

Till today various methods have been developed for mesh quality control. Generally, they can be classified in three main groups: the grid generation methods, the mesh smoothing methods and the mesh motion methods. The former group concerns about the generation of a smooth mesh within a 2D or 3D boundary [HDZ05, HX96, TSW99]. The second group of mesh smoothing methods im-

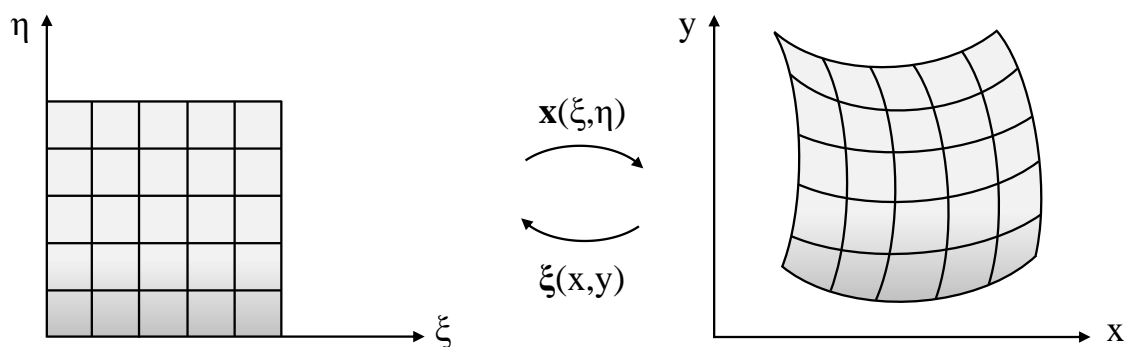
proves the quality of an existing 2D or 3D mesh [OBB01, Knu00, HZGB04], whereas the goal in the mesh motion methods is to find a valid 2D or 3D mesh after the movement of a 1D or 2D boundary [BSB07, JT06, SMA09]. The different approaches can be based on geometrical or mechanical considerations as in all fields the ideas are similar. Geometrical methods solve the mesh optimization problem locally since they are based on a local criterion, whereas mechanical methods use an auxiliary mechanical model to improve the mesh globally. Figure 2.25 gives an overview of the described classification of the different mesh quality control methods.



**Figure 2.25:** Classification of mesh quality control methods

### Grid generation

Numeric grid generation originally arose from the need to compute solutions to the partial differential equations of fluid dynamics for complex geometries [KS93]. The idea behind is to reduce the complexity of the meshing by separating the complicated shape of the physical region where the computation needs to be performed from the problem. Instead, the grid is generated on a simpler domain called the logical region and is transformed afterwards. This region can often be reduced to an unit square in two dimensions or an unit cube in three dimensions. Having generated a trivial grid in this area or volume, the goal is to define a transformation from the logical to the physical region also called map as shown in figure 2.26. In the two-dimensional case the variables  $x$  and  $y$  are the coordinates of the physical region while  $\xi$  and  $\eta$  are the coordinates of the logical one. A key element of a transformation is to avoid zero Jacobian



**Figure 2.26:** Transformation  $\mathbf{x}(\xi, \eta)$  from the logical (left) to the physical region (right) and the inverse transformation  $\xi(x, y)$

since this generates folded elements. Thus the problem of grid generation can be reduced to the problem of finding a valid transformation map for which the spacing between elements varies smoothly and the angles between grid lines do not become too small.

As described above, the methods for this type of problem can be geometrically or mechanically motivated. The Delaunay triangulation [She02, LL86] and the octree methods [Mar09] are two representatives of the geometrical methods. The former one creates a mesh of triangles in 2D or tetrahedrals in 3D based on points defining a surface or a volume, respectively. The octree methods create a grid in a three-dimensional space by recursively subdividing it into eight octants (or four quads in the 2D case).

On the other hand there are the mechanical methods which solve an auxiliary mechanical model for the unknown transformation. According to [KS93] one can distinguish between hyperbolic, parabolic and elliptic grid generators. Among them, the elliptical ones are most commonly used since their interior grid is very smooth and insensitive to the boundary parametrization even for non-smooth boundary data. For the grid generation an additional partial differential equation has to be solved. The simplest one is the Amsden-Hirt [AH73] grid generator which is based on the Laplace equation

$$\nabla^2 \mathbf{x} = \mathbf{0}. \quad (2.70)$$

More precisely, the map between the square logical region and the physical one is defined for the different variables in two dimensions as

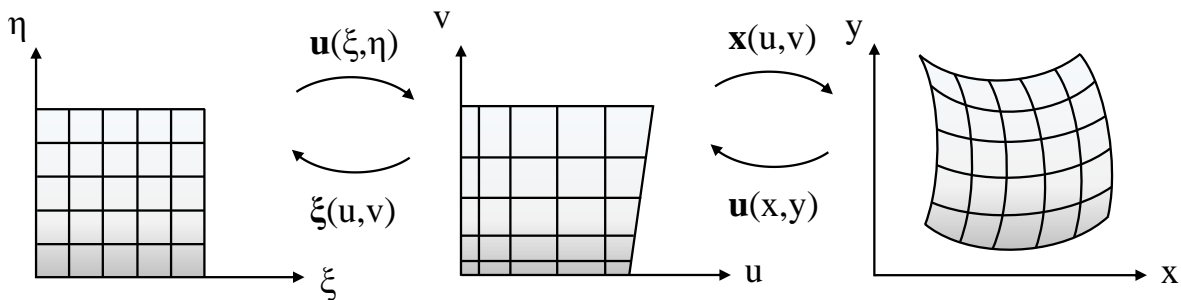
$$\begin{aligned}\nabla^2 x &= x_{\xi\xi} + x_{\eta\eta} = 0 \\ \nabla^2 y &= y_{\xi\xi} + y_{\eta\eta} = 0\end{aligned}\quad (2.71)$$

The Amsden-Hirt elliptic equation results in a linear system of equations and is solved in the logical domain. It has to be noted that for non-convex physical regions the transformation may have zero Jacobian which leads to folded meshes. For the same problem the Thompson-Thames-Mastin [TSW99] grid generator produces unfolded grids based on the inverse of equation (2.71):

$$\begin{aligned}\nabla^2 \xi &= \xi_{xx} + \xi_{yy} = 0 \\ \nabla^2 \eta &= \eta_{xx} + \eta_{yy} = 0\end{aligned}\quad (2.72)$$

The solution of this system of equations is performed in the physical region and in contrast to the Amsden-Hirt grid generator this leads to a nonlinear system of equations.

In order to gain a refined control over the grid, Castillo et al. [CSR87] introduced a third region, called the reference region, that already has some properties of the the physical domain but is usually considerably simpler as shown in figure 2.27.



**Figure 2.27:** Transformation for logical (left) to reference (middle) and physical region (right)

On the first sight grid generation seems not to be relevant in the context of node-based shape optimization as the optimization typically starts from a sufficiently smooth mesh which is preserved during the iterations. Nevertheless, the ideas

behind these methods are also adopted in the fields of mesh smoothing and mesh motion problems and therefore worth to mention in this work.

### Mesh smoothing

Regarding the class of mesh quality improvement based on geometrical approaches, Laplacian smoother are widely-used, e.g. in the field of computer graphics [OBB01] or for shape optimization of mechanical problems [LBT11]. The basic principle of the method is visualized in figure 2.28 where the mesh is smoothed by the movement of the node  $\mathbf{P}$ . In its simplest form, it repeatedly moves each mesh node by a displacement  $\mathbf{u}$  equal to a positive scale factor  $\gamma$  times the difference between the average of the neighboring  $n$  nodes  $\mathbf{x}_i$  and the node  $\mathbf{P}$ :

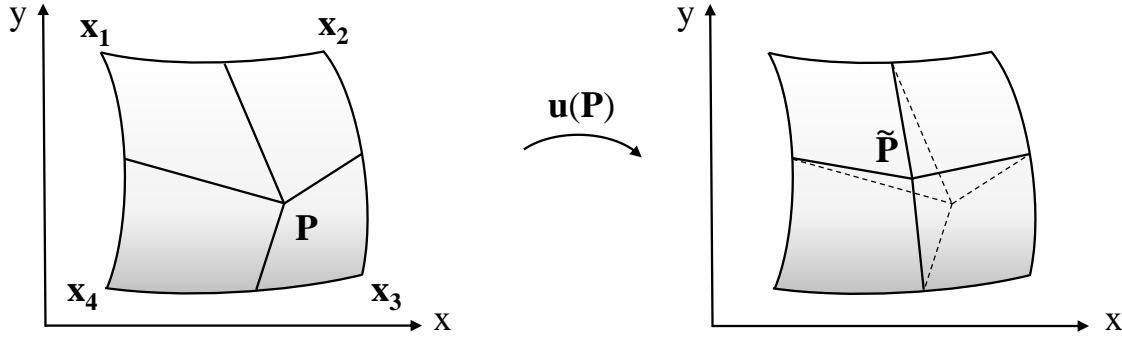
$$\mathbf{u}(\mathbf{P}) = \frac{1}{n} \cdot \sum_n \mathbf{x}_n - \mathbf{P} \quad (2.73)$$

$$\tilde{\mathbf{P}} = \mathbf{P} + \gamma \mathbf{u}(\mathbf{P}) \quad (2.74)$$

The nodal coordinates after the smoothing are denoted as  $\tilde{\mathbf{P}}$ . As a matter of fact, Laplacian smoothing is robust and requires only small numerical effort. However, it develops unnatural deformations when applied to highly irregular meshes thus several improvements have been sought. Taubin [Tau95] proposed to alternate two scale factors of opposite sign with the negative factor for larger magnitudes in the Laplacian smoothing algorithm. Such smoothing suppresses high frequencies of equation (2.73) while preserving and enhancing the low ones. Desbrun et al. introduced in [DMSB99] an accurate approximation of the mean curvature at the mesh vertex  $\mathbf{P}$  generating reliable and robust results even for irregular meshes.

Based on the work of [CSR87], Hansen et al. [HZGB04] proposed a method for the smoothing of unstructured grids. Herein, the quasi-linear elliptic Laplace equation (2.70) is discretized with Finite Elements and smoothing is achieved by computing a metric which incorporates the influence of neighboring elements.

A mechanical motivated mesh smoothing method is proposed by Stavropoulou et al. [SHB14, Sta15]. This approach is inspired by form-finding which is a



**Figure 2.28:** Mesh regularization by Laplacian smoother

method to determine the free-form equilibrium shape of a membrane subjected to a certain stress field [BR99, LB10, WB05]. Since no additional external body forces and surface tractions are acting, the weak form of equilibrium reduces to the internal work done by the predefined stress field,

$$\delta w(\mathbf{s}, \mathbf{r}) = \delta w_{int} = t \int_a \boldsymbol{\sigma}_0 : \delta \boldsymbol{\epsilon} da = 0. \quad (2.75)$$

Herein,  $\boldsymbol{\sigma}_0$  are the prescribed Cauchy stress tensor components acting on the resulting geometry with area  $a$ ,  $\boldsymbol{\epsilon}$  is the Euler-Almansi strain tensor [BW97] and  $t$  is the thickness which is considered to be constant. As the stress field is given no material definition is needed and the problem reduces to a purely geometrical one. But solving equation (2.75) directly the system of equations becomes singular with respect to the tangential movement  $\mathbf{r}$  because of the non-uniqueness of the discretization [BR99]. For the regularization Bletzinger et al. suggested in [BR99] the Updated Reference Strategy where the problem is solved in the reference configuration of the form-finding step  $i$ ,

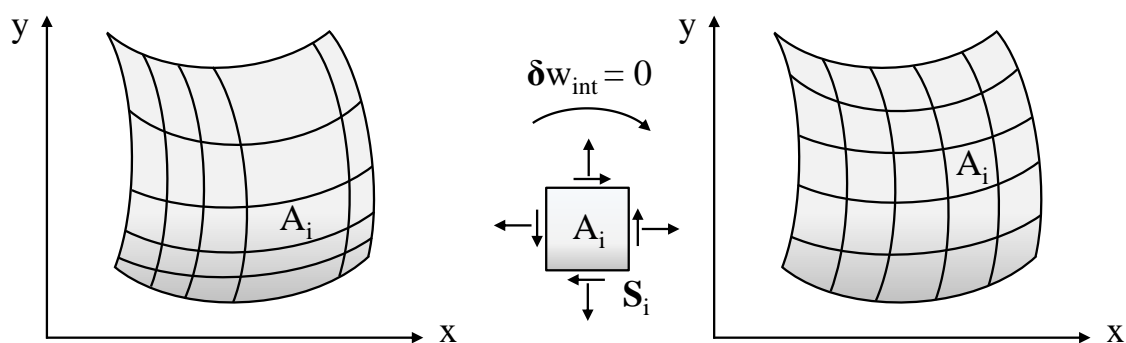
$$\delta w(\mathbf{s}, \mathbf{r}) = \delta w_{int} = t \int_{A^i} \mathbf{S}^i : \mathbf{E} dA^i = 0. \quad (2.76)$$

In this equation  $\mathbf{S}^i$  is the 2nd Piola-Kirchhoff stress tensor,  $\mathbf{E}$  is the Green-Lagrange strain tensor and  $A^i$  is the updated reference geometry of step  $i$ . But the difference to the original problem is that now the 2nd Piola-Kirchhoff stresses  $\mathbf{S}^i$  are assumed to be given instead of the Cauchy stresses  $\boldsymbol{\sigma}_0$  [Ble98] with the consequence that equation (2.76) is well defined and even linear with respect to  $\mathbf{s}$  and  $\mathbf{r}$ . At the end of the form-finding the Cauchy stresses  $\boldsymbol{\sigma}_0^i$  converge to the



predefined stress field  $\sigma_0$  and the final free-form shape is obtained.

For the mesh regularization the point of departure is again equation (2.75) but the idea is slightly modified: now the shape has to be retained and the discretization has to be improved. By the definition of a set of Dirichlet boundary conditions acting in normal to the surface direction the shape remains unchanged and only a movement of the nodes in the mesh relevant direction  $\mathbf{r}$  is possible. The desired mesh quality is achieved by the definition of an "ideal" element as reference configuration where the integration takes place and a prestress  $\mathbf{S}$  controlling the size or concentration of the elements. Figure 2.29 visualizes the described procedure.

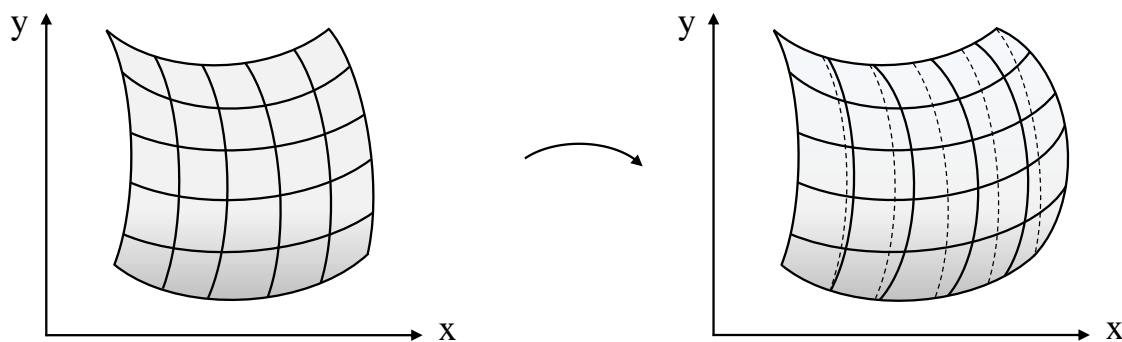


**Figure 2.29:** Mesh regularization by the Updated Reference Strategy with "ideal" element  $A_i$  and prestress  $\mathbf{S}_i$  (middle) for element  $i$

Generally, every element can have its individual "ideal" element, e.g. the initial high quality mesh at the beginning of an optimization or simply a unit square. The resulting system of equations is linear and can be solved in one step with respect to the tangential displacements  $\mathbf{r}$  for every node. In the special case of square initial "ideal" elements as template for all elements and an isotropic stress state, the method reduces to the aforementioned Amsden-Hirt method [Sta15].

### Mesh motion

According to figure 2.30 the third group of methods introduced in this section are the mesh motion methods. The goal herein is to find the surface or volume mesh after a 1D or 2D boundary is displaced. There is a broad variety of methods and the ideas are often similar to the ones for grid generation and mesh smoothing. Wang et al. [WCL18] and de Boer et al. [BSB07] developed a geometrically



**Figure 2.30:** Starting from an initial mesh (left) the goal is to find the resulting mesh after the boundary is displaced (right)

motivated approach which interpolates the displacements of the boundary nodes to the whole mesh with radial basis functions. In this method a small system of equations only involving the boundary nodes has to be solved without any additional information about the mesh connectivity. Apart from mesh motion problems, radial basis functions are well-established tools to interpolate data for example in the field of fluid-structure interaction where the discrete interfaces between the structural and the fluid mesh are often not matching [BW01, SCH00].

Farah et al. [DF02, FDKL98] proposed a mechanical approach based on a net of linear and torsional springs to transfer the displacements from the boundary to the whole domain. The method is successfully applied to update the mesh of fluid dynamic problems with moving boundaries. Another mechanical approach is presented by Jasak et al. in [JT06] where the diffusive properties of the elliptic Laplacian equation are used to compute the velocity  $\mathbf{u}$  of each mesh point,

$$\nabla(\gamma\nabla\mathbf{u}) = 0, \quad (2.77)$$

with a constant or variable diffusion field  $\gamma$ . Thus, the nodal coordinates  $\mathbf{x}_n$  of a point are evaluated from the previous position  $\mathbf{x}_{n-1}$  as

$$\mathbf{x}_n = \mathbf{x}_{n-1} + \mathbf{u}\Delta t, \quad (2.78)$$

where  $\Delta t$  is the time step. The boundary conditions for equation (2.77) are taken from the known boundary motion and the whole problem is solved using an

iterative linear equation solver [HS52]. Instead of using the diffusive properties of the Laplacian equation, Shimoda et al. [SMA09] and Riehl et al. [RFS<sup>+</sup>14] propagate the displacements of the boundary in the domain with the elastic properties of the continuum. In this method, the velocity field  $\mathbf{u}$  is obtained as a solution of a pseudoelastic problem in the pseudolinear elastic continuum defined on the actual design domain and loaded with a pseudodistributed external force  $\tilde{\mathbf{f}}$ , or traction, in proportion to the shape gradient function [AW96],

$$\tilde{\mathbf{K}} \cdot \mathbf{u} = \tilde{\mathbf{f}} \quad (2.79)$$

where  $\tilde{\mathbf{K}}$  denotes the pseudoelastic stiffness matrix. To account for invariable areas, a Dirichlet boundary condition is defined at these positions. The solution of the elliptic boundary value problem of equation (2.79) with respect to  $\mathbf{u}$  is called the Traction Method. The problem can be analyzed using any numerical analysis technique applicable to linear elastic problems, such as the Finite Element method. In order to further improve this method Azegami et al. [AT06] replaced the Neumann boundary condition  $\tilde{\mathbf{f}}$  with the Robin boundary condition which is a linear combination of the values of a function and the values of its derivative on the boundary of the domain. This advanced Traction Method provided smooth convergence to problems which could not be solved with the previous version.

The regularization method used in the Vertex Morphing method is also based on the elliptic Traction Method. To further improve the mesh quality conservation properties a variable Young's modulus being a function of the distance of an element from the design variables is incorporated in the pseudoelastic stiffness matrix  $\tilde{\mathbf{K}}$ . Moreover, to increase the flexibility of the mesh invariable areas are no longer completely fixed but can move freely in the tangential to the surface direction. Thereby, the mesh gains a higher level of flexibility with respect to the shape change. The details are presented in the next section.

### 2.6.5 Mesh update method

The proposed mesh regularization method is based on the Traction Method which is a method to smoothly update the Finite Element mesh of a domain due to displacement of the domain boundary [AT06][SMA09]. To further improve the method, the following extensions are implemented.

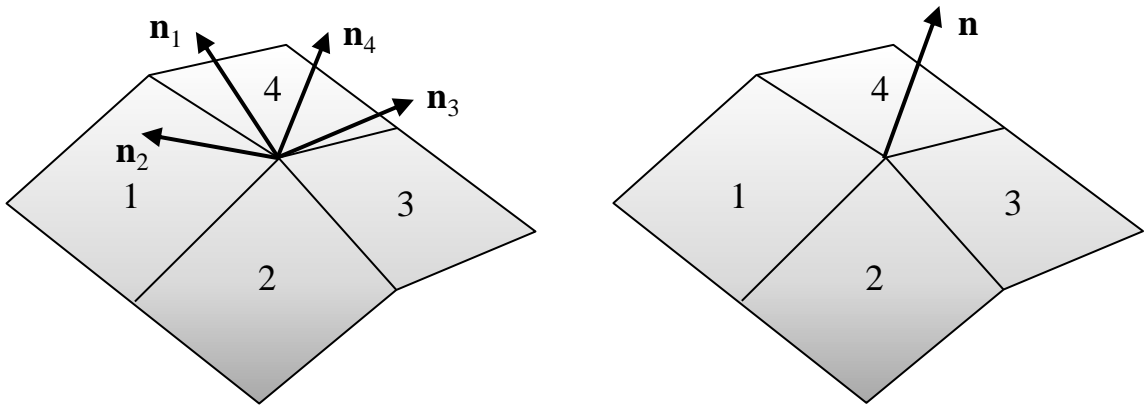
#### Definition of the boundary value problem

Beside the update of the Finite Element mesh in the whole domain, in node-based shape optimization problems the classical Traction Method is also used to smooth the raw sensitivity field  $\nabla_x J$  obtained by any sensitivity analysis method, e.g. adjoint method or Lagrange multiplier method. Thereby, the negative shape gradient  $-\nabla_x J$  is applied in the normal direction to the design boundary as an external traction force  $\tilde{\mathbf{f}}$ , i.e. Neumann condition to vary the shape. Using this method, the smoothed domain variation  $\mathbf{u}$  as the solution of equation (2.79) that minimizes the objective functional  $J$  can be obtained. The smoothness of the design update is assured by the smoothing properties of the elastic tensor  $\tilde{\mathbf{K}}$ .

According to section 2.6.3, in the Vertex Morphing method the regularization of the sensitivity field is already consistently incorporated in the optimization problem and consequently no additional smoothing is required anymore. Comparing both methods the smoothed velocity field  $\mathbf{u}$  at the design variables in the Traction method relates to the design update  $\delta\mathbf{x}$  in the Vertex Morphing method from equation (2.60). In order to simplify the control of the magnitude of shape change in an optimization iteration, the design update  $\delta\mathbf{x}$  is applied as Dirichlet boundary condition in equation (2.79). More precisely, for every design variable a local cylinder coordinate system is defined with the normal to the surface direction being the cylinder axis and the displacement is prescribed along this axis. Thus the change of the shape can be controlled by simply scaling  $\delta\mathbf{x}$  and the pseudoelastic problem reduces to

$$\tilde{\mathbf{K}} \cdot \mathbf{u} = \mathbf{0}, \quad (2.80)$$

Generally, equation (2.80) is defined in the global Cartesian coordinate system  $x, y, z$ . To restrict the prescribed displacement of the design variables to the normal to the surface direction, every variable  $s_l$  is transformed in its own local cylindrical coordinate system where the normal direction is defined as the cylindrical axis, cf. [Dho04]. This is the direction the design update  $\delta \mathbf{x}$  is applied. The remaining two directions, radial and tangential, are not constraint being part of the solution.



**Figure 2.31:** Normal direction before (left) and after the averaging (right)

In the Finite Element method the shape is not explicitly given and only an approximation of it can be estimated through the discretization. To avoid multiple normal directions as a consequence of the discretization an averaging step has to be introduced which is visualized in figure 2.31. The unit normal direction  $\mathbf{n}$  for every node can be computed as the average of the normals  $\mathbf{n}_i, i = 1, \dots, n_{ngh}$ , of the elements sharing the node as follows

$$\mathbf{n} = \frac{\sum_{i=1}^{n_{ngh}} \mathbf{n}_i}{\left\| \sum_{i=1}^{n_{ngh}} \mathbf{n}_i \right\|}. \quad (2.81)$$

In highly resolved meshes where the area  $A_i$  of the different elements has nearly the same size this approximation shows good results. In case of irregular meshes [Lin09] proposed a weighting with the inverse of the element area resulting in a

more accurate normal direction compared to equally weighted normals,

$$\mathbf{n} = \frac{\sum_{i=1}^{n_{ngh}} \frac{\mathbf{n}_i}{A_i}}{\left\| \sum_{i=1}^{n_{ngh}} \frac{\mathbf{n}_i}{A_i} \right\|}. \quad (2.82)$$

However, even this expression is just an approximation as discretization errors can not be avoided and in the course of the optimization an analytic description is not available anyway. But these inaccuracies are acceptable as they are automatically corrected during the optimization by the update of the design variables in normal direction.

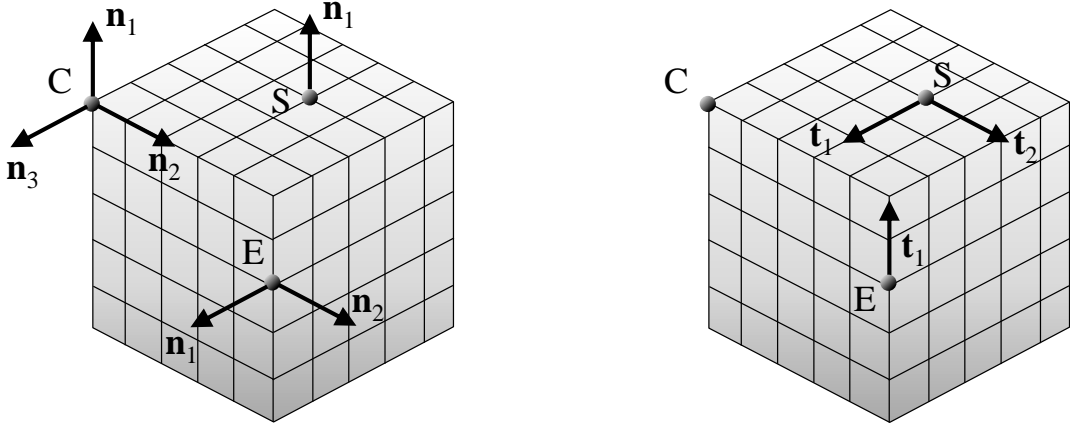
### Treatment of invariable areas

In the classical Traction Method [SMA09, AW96] invariable areas which are completely fixed in the so-called velocity analysis have to be defined manually in advance. Kernel of the extended Traction Method is the automated generation of proper boundary conditions for the surfaces of the mesh which are not part of the design space such that they do not undergo any design change but retain the highest possible flexibility with respect to the design update. The goal is to suppress all displacements which can alter the design (neglecting the discretization error) thus only a movement in tangential direction is possible. For the definition of these boundary conditions multiple point constraints were used. This is a very powerful concept to establish a relationship between different degrees of freedom [Dho18, Dho04] by eliminating a certain degree of freedom during the construction of the global stiffness matrix  $\tilde{\mathbf{K}}$ . As prerequisite the normal direction of the surface nodes of the invariable areas have to be assessed based on equation (2.81).

More precisely, according to figure 2.31 for every surface node the surface normal  $\mathbf{n}_k$  for each neighboring face  $k$  has to be calculated. For all possible combinations of faces, the scalar product

$$\alpha_{ij} = \frac{\mathbf{n}_i}{\|\mathbf{n}_i\|} \cdot \frac{\mathbf{n}_j}{\|\mathbf{n}_j\|}, \quad i \neq j, \quad i, j = 1, \dots, k, \quad (2.83)$$

has to be evaluated. Based on this and a predefined threshold for the scalar product, e.g.  $\alpha_{thresh} = 0.5$ , every surface node can be classified. Therefore, consider a discretized solid body as shown in figure 2.32. It is obvious that only



**Figure 2.32:** Surface characteristics of a solid body

a movement in normal direction  $\mathbf{n}_1$ ,  $\mathbf{n}_2$  and  $\mathbf{n}_3$  of a surface node will change the shape of the body. Movements along the surface tangent  $\mathbf{t}_1$  and  $\mathbf{t}_2$  will affect the mesh and not the shape. Thus, the displacements of surface nodes can either be "shape relevant" or "shape irrelevant" [BFF10] and three different types of nodes can be distinguished: surface nodes  $S$ , edge nodes  $E$  and corner nodes  $C$ . The latter ones are identified if the scalar product  $\alpha_{ij}$  for at least two pairs of normal directions  $\mathbf{n}_i$  and  $\mathbf{n}_j$  exceeds  $\alpha_{thresh}$ . These nodes are completely fixed in the mesh update step as every movement is shape relevant. All surface nodes  $S$  fulfill the requirement  $\alpha_{ij} < \alpha_{thresh}$ . This node type has only one shape relevant direction which has to be constrained in the construction of the global stiffness matrix  $\tilde{\mathbf{K}}$  by multiple point constraints. Therefore, let us assume that the global system of equations consist of  $N$  degrees of freedom

$$\sum_{j=1}^N a_{lj} u_j = b_l, \quad l = 1, \dots, N. \quad (2.84)$$

Additionally,  $M$  multiple point constraints are defined

$$\sum_{k=1}^N n_{ik} u_k = 0, \quad i = 1, \dots, M. \quad (2.85)$$

In our application the coefficients  $n_{ik}$  represent the corresponding spatial direction of the normal vector  $n$ . For each multiple point constraint  $i$ , one degree of freedom  $k_i$ , called dependent degree of freedom, is eliminated from the global system of equation (2.84). All dependent degrees of freedom  $k_i$ ,  $i = 1, \dots, M$  must be distinct. Based on Eq. (2.85) all the dependent degrees of freedom can now be collected at the left hand side:

$$\sum_{j=1}^M n_{ik_j} u_{k_j} = - \sum_{k=1, k \notin \{k_1, \dots, k_M\}}^N n_{ik} u_k, \quad i = 1, \dots, M. \quad (2.86)$$

This system of  $M$  equations and  $M$  unknowns can be solved for the dependent degrees of freedom resulting in the form

$$u_{k_j} = \sum_{k=1, k \notin \{k_1, \dots, k_M\}}^N c_{k_j k} u_k + d_{k_j}. \quad (2.87)$$

As we are concentrating on the surface nodes  $S$ , which have only one normal direction, only one multiple point constraint has to be considered, Thus, Eq. (2.87) reduces to

$$u_i = \sum_{k=1, k \neq i}^N c_{ik} u_k + d_i. \quad (2.88)$$

The displacement  $u_i$  can now be eliminated by substituting Eq. (2.88) into Eq. (2.84)

$$\sum_{j=1, j \neq i}^N (a_{lj} + a_{li} c_{ij}) u_j = b_l - a_{li} d_i, \quad l = 1, \dots, N. \quad (2.89)$$

The new coefficient  $\hat{a}_{lj}$  in the global stiffness matrix  $\tilde{\mathbf{K}}$  at position  $(l, j)$  now reads

$$\hat{a}_{lj} = a_{lj} + a_{li} c_{ij}, \quad j, l = 1, \dots, N; j \neq i. \quad (2.90)$$

An important property of the global stiffness matrix is its symmetry, which has a major advantage in the solution of the system of equations [Dho04]. Due to the operations above the symmetric structure is lost and has to be restored. Therefore, row  $i$

$$\sum_{j=1, j \neq i}^N (a_{ij} + a_{ii} c_{ij}) u_j = b_i - a_{ii} d_i \quad (2.91)$$



has to be multiplied by  $c_{im}$  and added to row  $m$ ,  $m = 1, \dots, N, m \neq i$ . The coefficient  $\hat{a}_{lj}$  at position  $(l, j)$  can now be expressed in the form

$$\hat{a}_{lj} = a_{lj} + a_{li}c_{ij}a_{ij}c_{il} + a_{ii}c_{ij}c_{il}, \quad j, l = 1, \dots, N; j, l \neq i. \quad (2.92)$$

The coefficient at the right hand-side also has to be modified:

$$\hat{b}_l = b_l - a_{li}d_i b_i c_{il} - a_{ii}d_i c_{il}, \quad l = 1, \dots, N; l \neq i. \quad (2.93)$$

Finally, row  $i$  and column  $i$  can be dropped from the global stiffness matrix, with the consequence, that for node  $S$  only degrees of freedom remain which allow a movement in direction of  $\mathbf{t}_1$  and  $\mathbf{t}_2$ . Edge nodes (see node  $E$  in figure 2.32) are characterized by exactly one pair of normal vectors  $\mathbf{n}_i$  and  $\mathbf{n}_j$  for which  $\alpha_{ij} > \alpha_{thresh}$ . This means that two shape relevant directions have to be constraint by multiple point constraints. In this way every shape relevant movement of every surface node which is not part of the design space is suppressed. For the sake of completeness, movements of nodes inside the body are always shape irrelevant in all three directions.

With these modifications of  $\tilde{\mathbf{K}}$  and the aforementioned displacements of the design variables, equation (2.80) can be solved for the vector  $\mathbf{u}$  which contains the smooth design update of the whole mesh. For the next iteration  $(i+1)$  the nodal coordinates of the Finite Element mesh are updated as follows

$$\mathbf{x}^{i+1} = \mathbf{x}^i + \mathbf{u}. \quad (2.94)$$

### 2.6.6 Mesh quality preservation

Apart from the mesh update, the preservation of a sufficient mesh quality during the optimization is of major interests. As aforementioned the smoothing properties of the linear elastic continuum globally retains a good mesh quality. But particular attention has to be payed to the elements in the vicinity of the design variables as the evolution of the shape mainly takes place there causing local mesh distortion and mesh dependencies in the gradient computation. The exam-

ple in section 2.6.1 revealed this fact clearly. To reduce this local deterioration an element dependent Young's modulus has been defined. Generally, the Young's modulus enters the global stiffness matrix

$$\tilde{\mathbf{K}} = \sum_{e=1}^{n_{el}} \tilde{\mathbf{K}}^e \quad (2.95)$$

with  $\tilde{\mathbf{K}}^e$  being the element stiffness matrix defined as

$$\tilde{\mathbf{K}}^e = \int_V (\mathbf{DN})^T \mathbf{E} \mathbf{DN} dV. \quad (2.96)$$

Herein,  $\mathbf{N}$  is the matrix of shape functions,  $\mathbf{D}$  is the matrix differentiation operator and  $\mathbf{E}$  is the elasticity matrix [Wri01]. In linear elastic problems  $\mathbf{E}$  has the following appearance:

$$\mathbf{E} = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix}, \quad (2.97)$$

where  $\lambda$  and  $\mu$  are the elastic Lamé constants which can be expressed in terms of the Young's modulus  $E$  and the Poisson's coefficient  $\nu$

$$\lambda = \frac{\nu E}{(1 + \nu)(1 - 2\nu)} \quad \mu = \frac{E}{2(1 + \nu)} \quad (2.98)$$

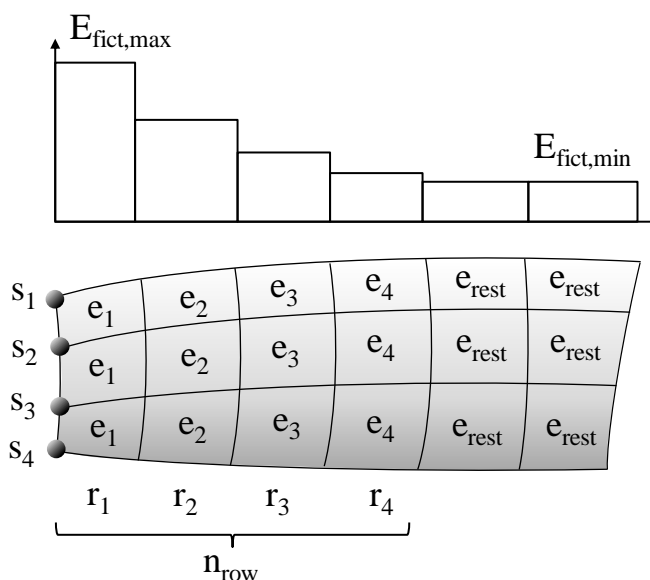
In order to prevent this deterioration a high value for  $E$  is assigned to elements in the vicinity of the design variables whereas distant elements receive a low one. The distribution of this fictitious Young's modulus is defined by an exponential decay function

$$E_{fict}(r) = E_{fict,max} \cdot e^{-\Delta E \cdot r}, \quad (2.99)$$

with the decay constant

$$\Delta E = \frac{\ln\left(\frac{E_{fict,max}}{E_{fict,min}}\right)}{n_{row}}. \quad (2.100)$$

The fictitious Young's modulus is a function of the element row  $r$  which is a measure for the distance of a specific element from the design variables  $\mathbf{s}$ , see figure 2.33. The values of the Young's modulus vary between the maximum  $E_{fict,max}$  and the minimum  $E_{fict,min}$ . The definition of the decay constant  $\Delta E$  ensures that after a predefined number of rows  $n_{row}$  the minimum Young's modulus is reached.



**Figure 2.33:** Variation of Young's modulus  $E$  assigned to different element rows

For the definition of the element sets  $e_i$  of the different rows  $r$ , the topology of the mesh has to be taken into account according to algorithm 1. At first the expression  $\Delta E$  has to be calculated as an input for the function  $E_{fict}(r)$ . Based on the available information about the mesh topology all elements which at least contain one design variable are put in the set  $e_1$  and receive the highest fictitious Young's modulus  $E_{fict}(r = 1)$ . For the subsequent element rows  $r > 1$  the steps (4) to (10) have to be carried out  $(n_{row} - 1)$  times. In step (6), all nodes of the elements from the previous row have to be collected. For these nodes all elements which contain at least one of them are put in the next set. According to step (8) the elements which already received a Young's modulus have to be

**Algorithm 1** Assessment of the fictitious Young's modulus for the different element rows

---

**Input:** Number of rows for scaling:  $n_{row}$

**Input:** Young's modulus:  $E_{fict,max}, E_{fict,min}$

**Output:** Young's modulus distribution:  $E_{fict}(r)$

- 1: Calculate  $\Delta E$  from equation (2.100)
  - 2: Find all elements  $e_1$  containing at least one design variable  $s$
  - 3: Assign Young's modulus  $E_{fict}(r = 1)$  to the set  $e_1$
  - 4: **if**  $n_{row} > 1$  **then**
  - 5:     **for**  $r = 1$  **to**  $(n_{row} - 1)$  **do**
  - 6:         Determine all nodes  $s_r$  of the elements in  $e_r$
  - 7:         Determine all elements  $e_{r+1}$  containing at least one node from set  $s_r$
  - 8:         Subtract elements from sets already treated  $e_{r+1} = e_{r+1} - \sum_{i=1}^r e_i$
  - 9:         Assign Young's modulus  $E_{fict}(r + 1)$  to set  $e_{r+1}$
  - 10:     **end for**
  - 11: **end if**
  - 12: Assign Young's modulus  $E_{fict,min}$  to all other elements ( $e_{rest}$ )
- 

deleted from this set and the Young's modulus can be calculated. In the last step the Young's modulus  $E_{fict,min}$  is assigned to the remaining elements.

Due to the scaling two positive effects can be observed: Elements in the vicinity nearly behave like rigid bodies, which reduces the local mesh deformation and mesh dependencies during the optimization. Especially in case the stresses are optimized, a sufficient mesh quality near the design variables is essential for the computation of reliable local stresses and gradient information. The second positive effect is that the design update is distributed more even in the whole mesh. Consequently more elements are involved in the shape update process reducing the risk of a fast deterioration of single elements within a few design iterations.

# Optimization workflow

---

In chapter 2 the theoretical background and the different elements necessary for a parameter-free, gradient-based shape optimization process were discussed. As a key element the adjoint semi-analytical sensitivity analysis is introduced in section 2.2. Important response functions and geometric constraints for structural optimization problems are discussed in section 2.3 and 2.4. Section 2.5 concentrates on the improvement of the gradient field with respect to the dependency of the mesh topology and the element formulation. The update of the discretized design and the conservation of the mesh quality are presented in section 2.6. These are the basic elements for the proposed optimization process. In this chapter the focus is on the combination of these single modules to a complete workflow, the organization of the data structure between the different elements and some implementation issues. Furthermore, the aggregation of constraints is discussed for the development of feasible designs for constraint optimization problems.

Thus, this chapter is organized as follows: Section 3.1 introduces the overall formulation for structural optimization problems and introduces the general optimization workflow. The different elements of this workflow are presented in detail in the next sections. More precisely, in section 3.2 the optimization algorithms for constrained and unconstrained optimization problems are described. In section 3.3 the update of the design in every iteration is discussed and in 3.4

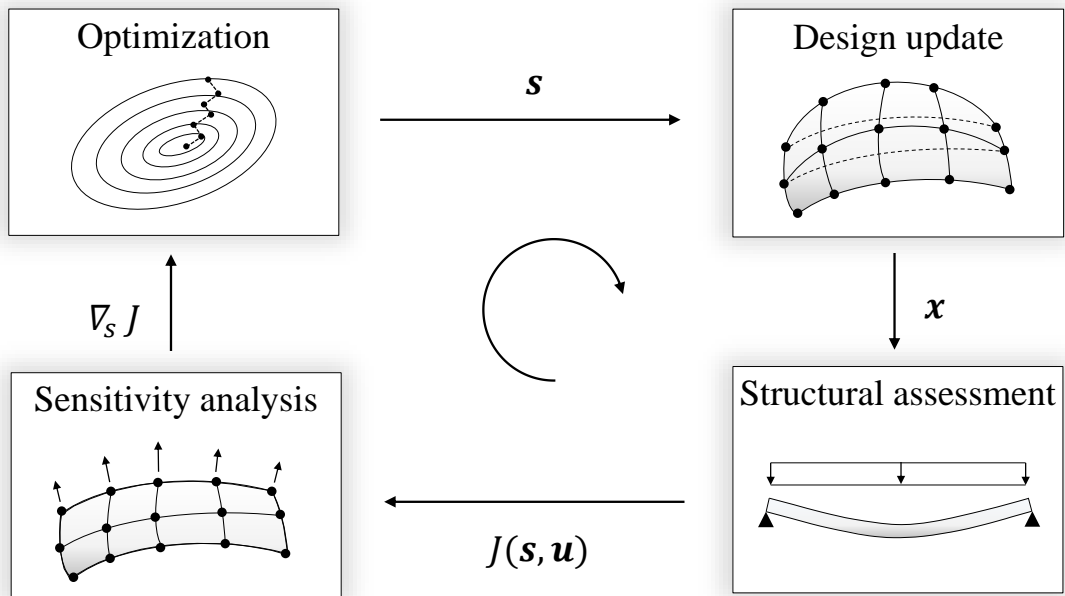
the structural assessment and the corresponding sensitivity analysis is addressed. Finally, implementation related topics are discussed in the last section 3.5.

## 3.1 Computational framework

In the literature, basically two main approaches for the formulation of structural optimization problems are commonly used. The first one is called "nested analysis and design" (NAND) and the second one is the "simultaneous analysis and design" (SAND) approach. In the SAND formulation which is discussed since the 1960s the state variables which are typically the displacements in structure mechanical problems are also treated as design variables and the partial differential equations describing the physical system are considered as an additional equality constraint in the optimization problem. The basic idea behind is to transform an constrained optimization problem in the design variable space into an unconstrained problem in the mixed space of design and state variables. In addition, since the physical system is embedded in the optimization problem no explicit structural or sensitivity analysis is needed. It is also important to note that in the SAND approach the equilibrium equation does not necessarily has to be fulfilled in every optimization iteration. In contrast, in the NAND approach the design variables are treated as independent optimization variables. The result of the physical problem, such as stresses and strains, depends implicitly on the design variables. Therefore, the equilibrium equations have to be satisfied in every iteration. For a deeper understanding about the details of both formulations the reader is referred to [AW05].

In this work, the NAND approach is applied and the corresponding parameter-free, gradient-based shape optimization workflow is presented in figure 3.1. The optimization problem is solved in an iterative manner meaning the elements of the workflow are repeatedly executed until convergence. Initially, in the structural assessment the solution of the underlying physical problem is provided which is the basis for the corresponding sensitivity computation. Then, according to figure 3.1, the optimizer suggests the update of the design variables. Subsequently the whole mesh is updated according to the prescribed displacements of the design

variables. On the basis of this new geometry, the physical equations are not in equilibrium anymore thus the state variables have to be solved again and the next iterations starts. This workflow is repeated until a certain convergence criterion is met, e.g. change of the response function. In the next sections each of these elements is described in detail.



**Figure 3.1:** Main elements of the optimization workflow

## 3.2 Optimization algorithm

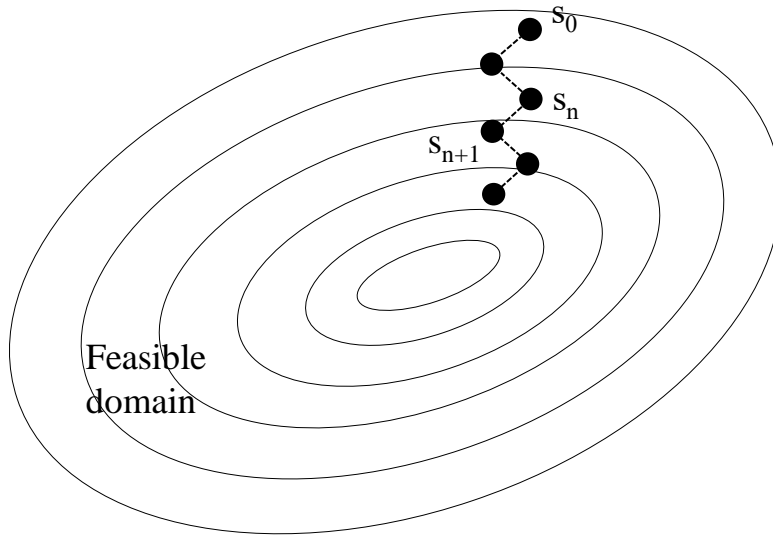
In the computational framework the optimizer is responsible for the update of the design variables based on the gradient information computed in the sensitivity analysis step. In practice, one can distinguish between zero-order, first-order and second-order gradient information methods. A broad and detailed overview of the different optimization algorithms is given in [Kir92, Rao09]. In this work first-order methods are applied namely the steepest descent method for unconstrained optimization problems and Rosen’s gradient projection method [Ros60, Ros61] for constrained optimization problems.

### 3.2.1 Unconstrained optimization

The shape optimization process can be considered as the evolution of the design  $\mathbf{s}$  in several small increments  $n$  as follows:

$$\mathbf{s}_{n+1} = \mathbf{s}_n + \alpha_n \mathbf{d}_n. \quad (3.1)$$

According to this equation and figure 3.2, the update of the design variables  $\mathbf{s}_{n+1}$  in every iteration is based on the previous design  $\mathbf{s}_n$  consequently requiring an initial guess  $\mathbf{s}_0$  in the first iteration. The design change itself is achieved by the modification of  $\mathbf{s}_n$  with the product of the search direction  $\mathbf{d}_n \in \mathbb{R}^n$  and a positive step size  $\alpha_n$ .



**Figure 3.2:** Design update with Steepest Descent algorithm for an unconstrained optimization problem

Generally, a gradient vector  $\mathbf{d}_n$  can be considered as suitable descent direction if the condition

$$(\nabla_s J(\mathbf{s}_n))^T \mathbf{d}_n < 0 \quad (3.2)$$

is fulfilled guaranteeing an angle greater than  $90^\circ$  between the gradient of the response function  $\nabla_s J(\mathbf{s}_n)$  and  $\mathbf{d}_n$ . As  $\nabla_s J(\mathbf{s}_n)$  points in the direction of the steepest ascent  $\mathbf{d}_n$  has to be a direction in which the function value decreases. The computation of the descent direction can be generalized as

$$\mathbf{d}_n = -\mathbf{D}_n \nabla_s J(\mathbf{s}_n), \quad (3.3)$$



where  $\mathbf{D}_n$  is a positive definite matrix. Replacing  $\mathbf{D}_n$  by the inverse of the Hessian  $(\nabla_s^2 J(\mathbf{s}_n))^{-1}$  results in the well-known Newton's method commonly applied for the solution of nonlinear optimization problems. However, the large number of design variables makes the approximation or even evaluation of the Hessian very expensive [Hoj14] being the reason why this approach is not very common in practice. If  $\mathbf{D}_n$  is defined as the identity matrix  $\mathbf{I}$ , the expression reduces to the Steepest Descent method where

$$\mathbf{d}_n = -\nabla_s J(\mathbf{s}_n). \quad (3.4)$$

In 1847 Cauchy was the first who observed the properties of the negative gradient of the response function being exactly the direction where the value of the response function decreases the fastest. Thus, this method may seem to be the best algorithm since every one-dimensional search starts in the direction of the steepest descent [Rao09]. This statement holds for points far away from the minimum but in the vicinity of the minimum it converges rather slowly. Nevertheless, this algorithm has been chosen for several reasons.

First, the Vertex Morphing method introduces an additional design space where the optimization problems is solved and linked to the geometry by a kernel filter. As derived in section 2.6 this approach is equivalent to the implicit Sobolev-gradient smoothing which is already enhanced by some approximation of the Hessian matrix.

Furthermore, from a geometrical point of view, the design variables are defined on a local surface coordinate system approximated by the normal in every node. Due to this reason the coordinate system varies in every optimization iteration which would require the storage of the deformation gradient for every surface node in every step if higher-order algorithms are used. The steepest descent algorithm makes the whole optimization history independent and simple.

Beside the descent direction, the step size  $\alpha_n$  is the second variable driving the design update and having a major influence on the convergence of the

optimization problem. Ideally  $\alpha_n$  is determined by minimizing the function  $J(\alpha) = J(\mathbf{s}_n + \alpha_n \mathbf{d}_n)$  with respect to  $\alpha_n$  which is called exact line search. Due to the fact that this approach becomes very costly even for simple optimization problems, it has only minor practical relevance. Here, the initial step length  $\alpha_0$  is predefined based on the geometric dimensions of the initial design and automatically bisected towards the minimum according to the Armijo test algorithm in combination with a backtracking line search [LY08]. The procedure is shown in detail in algorithm 2.

---

**Algorithm 2** Line search algorithm

---

**Input:** coefficients:  $\tau \in (0, 1), \rho \in (0, 1)$

**Output:** actual step length:  $\alpha_n$

- 1: **while**  $J(\mathbf{s}_n + \alpha \mathbf{d}_n) > J(\mathbf{s}_n) + \tau \alpha (\nabla_s J(\mathbf{s}_n))^T \mathbf{d}_n$  **do**
  - 2:      $\alpha = \rho \alpha$
  - 3: **end while**
  - 4: set  $\alpha_n = \alpha$
- 

As described in the above algorithm two coefficients have to be defined as input parameters, namely  $\tau$  which defines the amount of reduction proportional to the step length and the search direction and  $\rho$  which reduces the step length. Typically,  $\tau$  is chosen rather small, e.g.  $1.0 \times 10^{-5}$ , and  $\rho$  is set to 0.5. In the "while loop"  $\alpha$  is bisected until the inequality is fulfilled which states that the response function of the new design has to be lower than the old design plus a portion of the gradient vector.

This approach belongs to the group of "inaccurate" line search algorithms as inaccuracy is introduced by simply terminating the search procedure before it has converged. However, in practice this is often acceptable in order to conserve overall computation time [LY08].

### 3.2.2 Aggregation of constraints

Typically in real industrial applications constraints are immanent, e.g. the maximum admissible mass of a component or the limitation of the available space. These constraints divide the design space into two domains, the feasible one

where the constraints are satisfied and the infeasible one where at least one constraint is violated. For the various algorithms to solve such constrained optimization problems one can refer to Bertsekas [Ber99] and Luenberger et al. [LY08]. In this work Rosen's Gradient Projection method [Ros60, Ros61] is used due to the advantageous computational efficiency of CalculiX and the straightforward integration in the optimization strategy as explained later in the section.

The method is motivated by the ordinary Steepest Descent algorithm described above. The main idea is to project the function gradient on the subspace tangent of all active constraints in order to find the direction of movement. The process is always initiated at a feasible design  $\mathbf{s}_n$  where a certain number of active constraints  $G_i, i = 1, \dots, n_{act}$ , exist. At this point of time a feasible direction  $\mathbf{d}_n$  is sought fulfilling equation (3.2) so that the response function  $J$  is reduced as follows:

$$\mathbf{d}_n = -\nabla_s J(\mathbf{s}_n) \mathbf{P} = \mathbf{P}(-\nabla_s J(\mathbf{s}_n)) \quad (3.5)$$

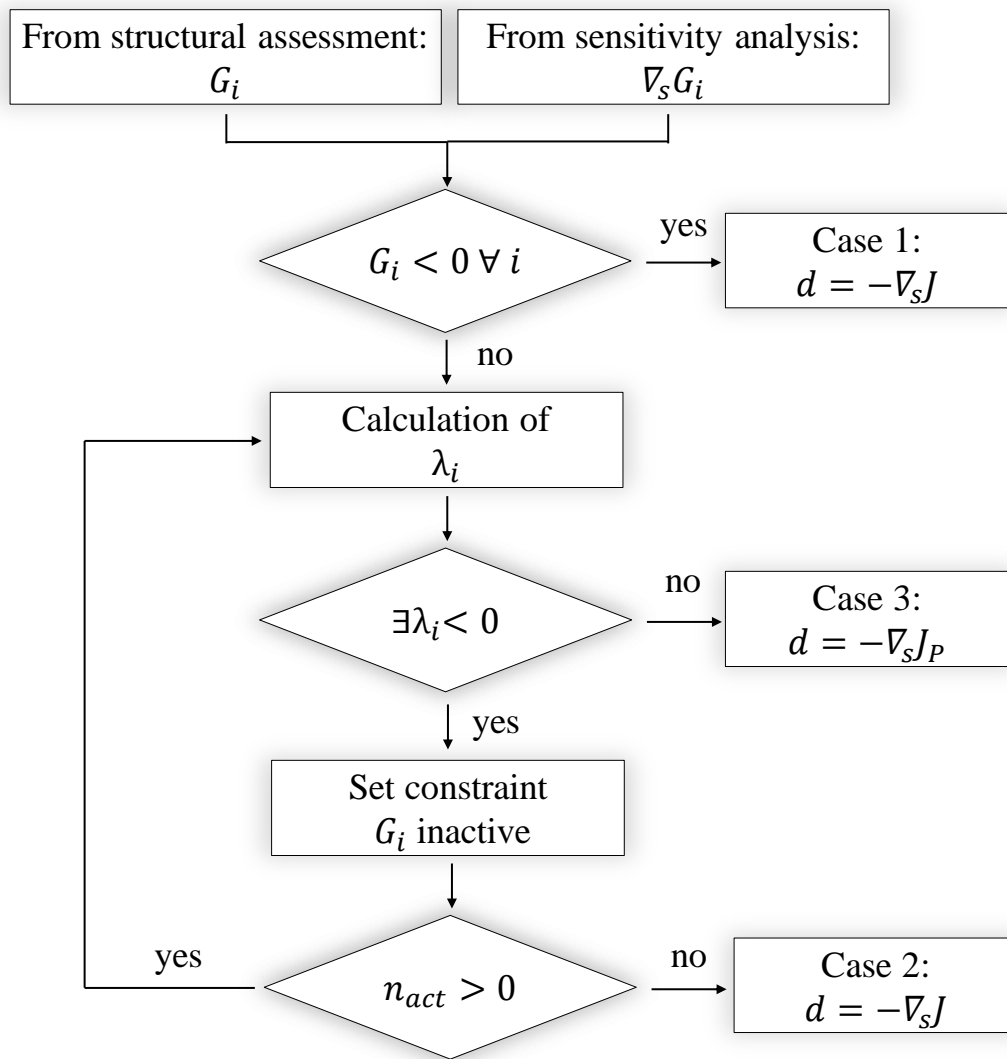
The projection matrix  $\mathbf{P} \in \mathbb{R}^{n \times n}$  is computed based on the following matrix operation

$$\mathbf{P} = \mathbf{I} - \mathbf{N}(\mathbf{N}^T \mathbf{N})^{-1} \mathbf{N}^T \quad (3.6)$$

with  $\mathbf{I}$  being the identity matrix and  $\mathbf{N}$  being a matrix of rank  $n \times n_{act}$  which contains the gradient information of all active constraints.

$$\mathbf{N} = [\nabla_s G_1(\mathbf{s}_n), \nabla_s G_2(\mathbf{s}_n), \dots, \nabla_s G_{n_{act}}(\mathbf{s}_n)] \quad (3.7)$$

The projection matrix has to be computed if at least one constraint is active based on the algorithm shown in figure 3.3. As initial information the function values  $G_j$  and the gradient vectors  $\nabla_s G_j(\mathbf{s}_n)$  have to be computed for all inequality constraints  $j = 1, \dots, n^G$  and the lower and upper bounds (the corresponding function value is one and the gradient vector is introduced in section 2.4). In the first step the validity of equation (2.1) is checked and the constraints violating it are added to the set of active constraints  $n_{act}$ .



**Figure 3.3:** Algorithm for the assessment of a feasible descent direction in consideration of constraints

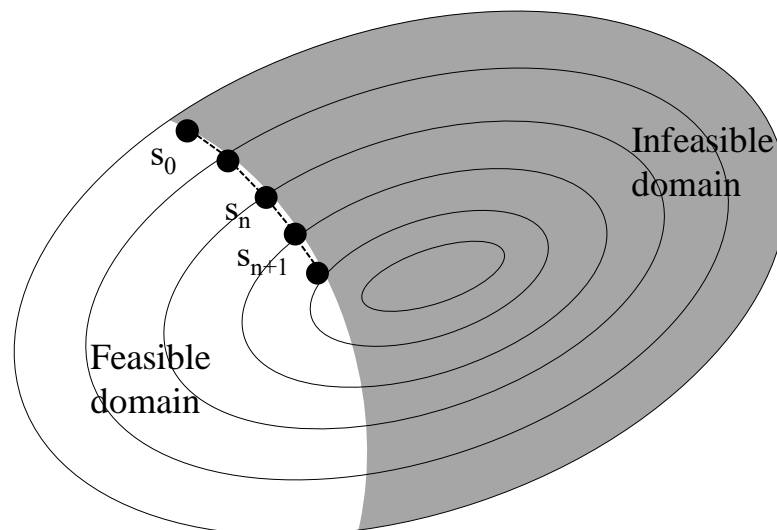
In case the number of active constraints  $n_{act}$  equals zero, the algorithm is terminated indicating that an unconstrained optimization problem is present. Figure 3.2 visualizes this situation where the actual design point lies inside the feasible domain and the descent direction is simply the negative gradient from equation (3.4).

If the number of active constraints  $n_{act} > 0$ , the corresponding Lagrange multipliers have to be computed,

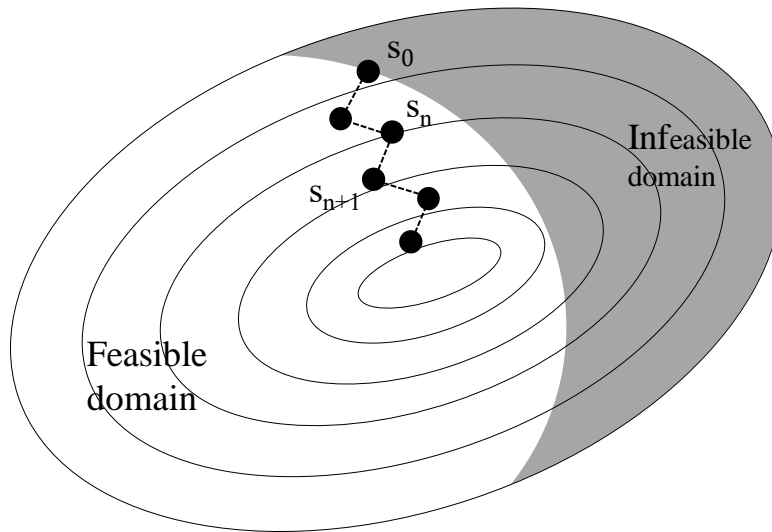
$$\lambda = -(\mathbf{N}^T \mathbf{N})^{-1} \mathbf{N}^T \nabla_s J(\mathbf{s}_n). \quad (3.8)$$

The case all the components of  $\lambda$  are non-negative implies that the Karush-Kuhn-Tucker conditions for the original problem are satisfied at  $s_n$  and the projection matrix can be computed. As shown in the corresponding figure 3.4, the steepest descent direction to the minimum is restricted by the active constraints and the descent direction  $\mathbf{d}$  has to follow the constraint boundary until no further improvement of the objective function is possible or a constraint becomes inactive.

If, however, at least one component of  $\lambda$  is negative, it is possible, by relaxing the corresponding inequality, to move in a new direction. A negative value  $\lambda_i$  implies that the current design indeed lies on the boundary of  $G_i$  but this constraint is not active as the gradient vector  $\mathbf{d}$  points in direction of the feasible domain. This case is described in figure 3.5 where the initial design  $s_0$  lies on the boundary of a constraint but the shape update happens inside the feasible domain and does not violate the constraint. Consequently  $G_i$  can be eliminated from the set of active constraints  $n_{act}$ . Following the proposed algorithm, the Lagrange multipliers have to be re-calculated for the reduced set and checked again for negative elements.



**Figure 3.4:** Design update where the initial design lies on the boundary of a active constraint



**Figure 3.5:** Design update where the initial design lies on the boundary of a non-active constraint

Generally, the techniques of linear and nonlinear programming are of greatest use in connection with a large system involving many design variables and being subjected to constraint equations [Ros60]. Rosen's gradient projection method belongs to this group of methods. The original constrained optimization problem is transformed into a sequence of matrix and vector multiplications and the computation of the inverse of  $(\mathbf{N}^T \mathbf{N})^{-1}$  needed in equation (3.6) and (3.8). But instead of computing this inverse directly, it is obtained by solving  $n_{act}$  system of equations which is from a numeric view point more stable. Furthermore, the sequence of the mathematical operations is arranged such that the storage of the projection matrix can completely be avoided. All these operations can efficiently be implemented in CalculiX on the basis of the existing routines and are shown in detail in Appendix B.

Furthermore, the gradient projection fits very well in the concept of updating the design by the product  $\alpha_n \mathbf{d}_n$  due to the fact that on the subspace tangent the method reduces to the simple Steepest Descent method for unconstrained optimization problems.

### 3.3 Design update

The next step after the determination of the improved set of design variables  $\mathbf{s}_{n+1}$  by the optimizer is the update of the shape. This step has to fulfill several tasks, namely the update of the mesh according to the value of the design variables while keeping all other surfaces unaltered, update the whole volume mesh and retain the quality of the finite elements. As described in section 2.6.3, in the Vertex Morphing method the update of the surface geometry is linked with the update of the design variables through the following expression:

$$\delta \mathbf{x} = \mathbf{A} \delta \mathbf{s} \quad (3.9)$$

The geometry update is the main input for the proposed Traction method (section 2.6.5) used for updating the whole volume mesh in order to conform to the new boundary geometry. Therefore, the smoothing properties of the global stiffness matrix in the linear elastic problem

$$\tilde{\mathbf{K}} \cdot \mathbf{u} = \mathbf{0} \quad (3.10)$$

are used to distribute the surface displacements homogeneous in the volume mesh. Further improvements of the method are automatically provided by CalculiX according to the proposed method in the sections 2.6.5 and 2.6.6: first, the multiple point constraints which improve the flexibility of the invariable surface mesh by allowing a movement tangential to the surface. Second, The variable Young's modulus distribution further reducing mesh dependencies and mesh degeneration.

Having all the input, the optimization algorithm automatically creates a new Finite Element model which is solved in a linear elastic analysis for the unknown nodal displacements  $\mathbf{u}$ . The update of the whole volume mesh - derived from the update of design variables - is finally transferred to the structural assessment.

## 3.4 Structural assessment and sensitivity analysis

The free Finite Element program CalculiX (the GNU General Public License applies) offers many different methods for diverse physical areas of applications, e.g. static and dynamic analysis, CFD problems, or eigenvalue problems. The whole spectrum of available methods is shown in figure 3.6. The parameter-free shape optimization process presented in this work is destined to optimize linear and nonlinear (considering geometric and material nonlinearity) static problems.

In the course of this research project a semi-analytical adjoint sensitivity analysis module has been implemented in CalculiX, see method twelve in figure 3.6. This step always starts directly after a static assessment in the same CalculiX run since it needs the information from the previous step for the computations of the gradients like the global stiffness matrix, the stress state or the displacement field. Both the structural assessment and the sensitivity analysis have to be reevaluated in every optimization iteration after the change of the geometry since these results are required by the Steepest Descent algorithm.

CalculiX

1. Static analysis	2. Frequency analysis	3. Buckling analysis
4. Dynamic analysis	5. Steady state dynamic analysis	6. Coriolis frequency calculation
7. Flutter frequency calculation	8. Magnetostatics	9. Magnetodynamics
10. Electromagnetic eigenvalue problem	11. Superelement creation	12. Sensitivity analysis

**Figure 3.6:** Available methods in the free Finite Element code CalculiX

Regarding the time consumption, for the applications presented in the next chapter the sensitivity analysis consumes most of the computational time. Depending on the number of CPU's the solution of the static problem is by a factor two up



to six faster than the computation of the gradients. However, this experience can not be applied to arbitrary models since the ratio between the structural assessment and the sensitivity analysis depends on various parameters, e.g. the overall number of degrees of freedom of the Finite Element model, the number of design variables, the size of the sets for the response functions and the degree of parallelization. Therefore, a detailed time consumption study is presented in section 4.3.2.

## 3.5 Implementation aspects

The presented design process provides the software platform for the structural shape optimization of large problems. A discretized structural case and an additional optimization inputdeck specifying the optimization task is given to the workflow. In the course of the optimization a part of the domain surface defined by the design variables is altered to improve the objective function. As final result an updated discretization of the underlying Finite Element model is submitted by the workflow.

This whole optimization workflow is defined in a modular way which eases the exchange of specific modules or their adaption for other applications. The workflow as well as the interaction of the different modules is presented in figure 3.7. Herein, the following two software packages are used: The open source Finite Element program CalculiX [Dho18] applied for the solution of various physical problems and the programming language Python [Van17].

The latter one is used for two different tasks. First, the whole optimization workflow is embedded in the Python environment which is responsible for the data exchange between the different modules and for the driving the optimization. As a second application, the step length control module is also realized in Python containing the Armijo test algorithm and the Backtracking line search. Herein, the step length parameter  $\alpha$  is determined on basis of the descent direction and the response function values.

The Finite Element program CalculiX is the kernel of the whole optimization workflow. The program is written in C and Fortran using the advantages of both languages. The C routines build the framework of the program being responsible for the memory allocation, the variable definition and the solution of the systems of equation. The computational tasks are mainly done in the Fortran routines due to its intuitive programming language. Generally, CalculiX is capable to solve various physical problems as figure 3.6 demonstrates. But here only the linear and nonlinear structural problems are of interest. As aforementioned, the semi-analytic adjoint sensitivity analysis method has been implemented in CalculiX in the course of this research project. Beside the gradient computation it also contains the out-of-surface regularization, the sensitivity weighting and the Gradient Projection computation for constrained problems. Both the structural assessment and the sensitivity analysis form another module in the workflow. The advantage of combining both steps in one tool is that the information generated in the static step and needed in the sensitivity step can easily be passed. Notice that this module is freely available in CalculiX ([www.dhondt.de](http://www.dhondt.de)).

Having computed the update of the design variables, the volume mesh is adapted by the solution of a linear elasticity problem with a non-uniform Young's modulus distribution and multiple point constraints on all boundaries of the domain. In this mesh motion module again CalculiX is used. The displacements of the design variables are distributed through the elastic properties of the global stiffness matrix assembled with a Young's modulus proportional to the inverse of the distance to the design variables. This results in an more evenly deformed mesh which is the basis for the next optimization iteration. The automated generation of the multiple point constraints is also freely available in CalculiX.

Regarding some further implementation aspects, the structural assessment as well as the sensitivity analysis enable a parallel computation on several CPU's as these are the most time-consuming parts of the workflow. Thus, the sequential parts of the codes are clearly reduced. This means that the contribution to matrices and vectors of system size are assembled element-wise on the sub-domain of the single processor whenever advantageous. The efficiency of the parallelization

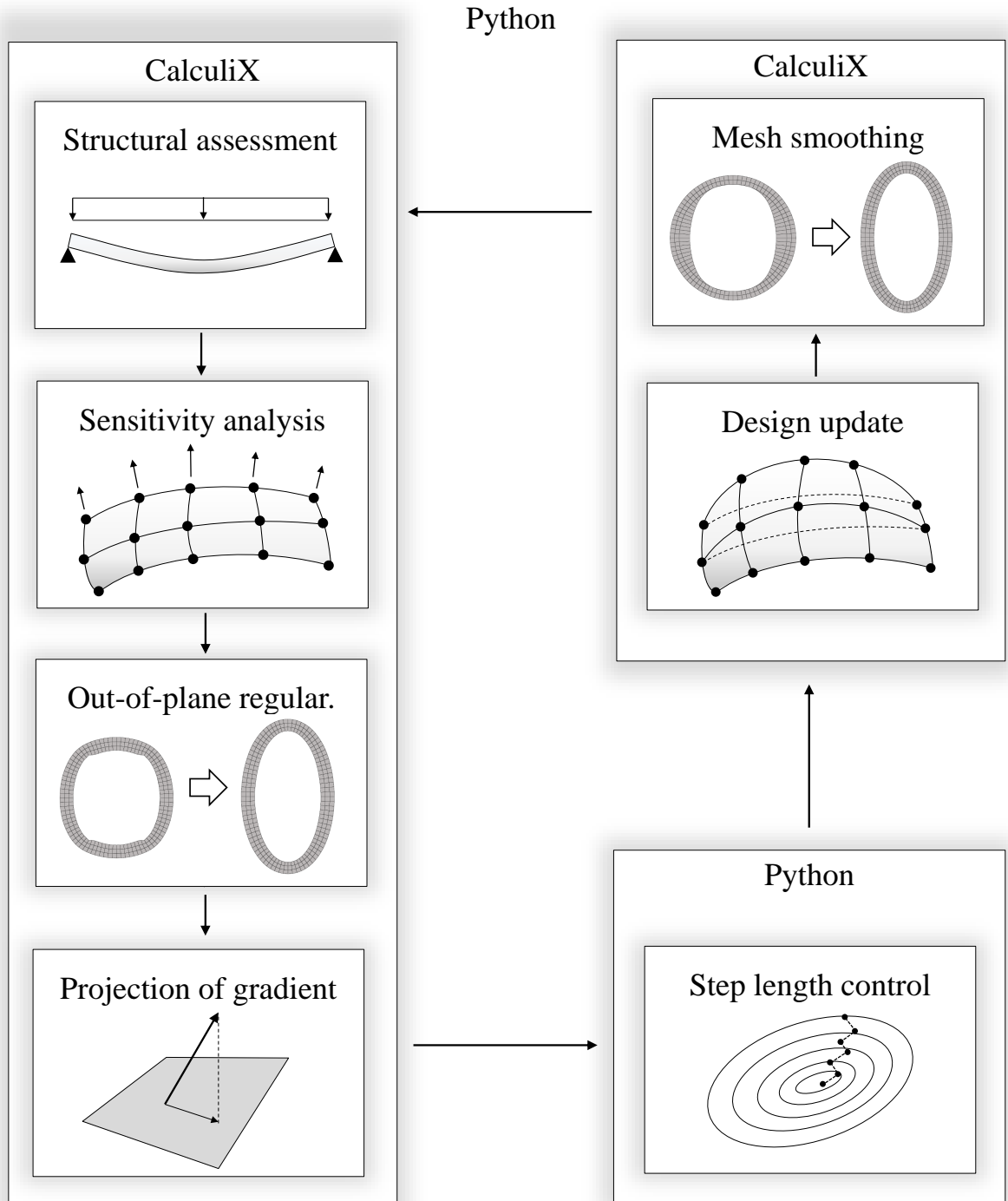
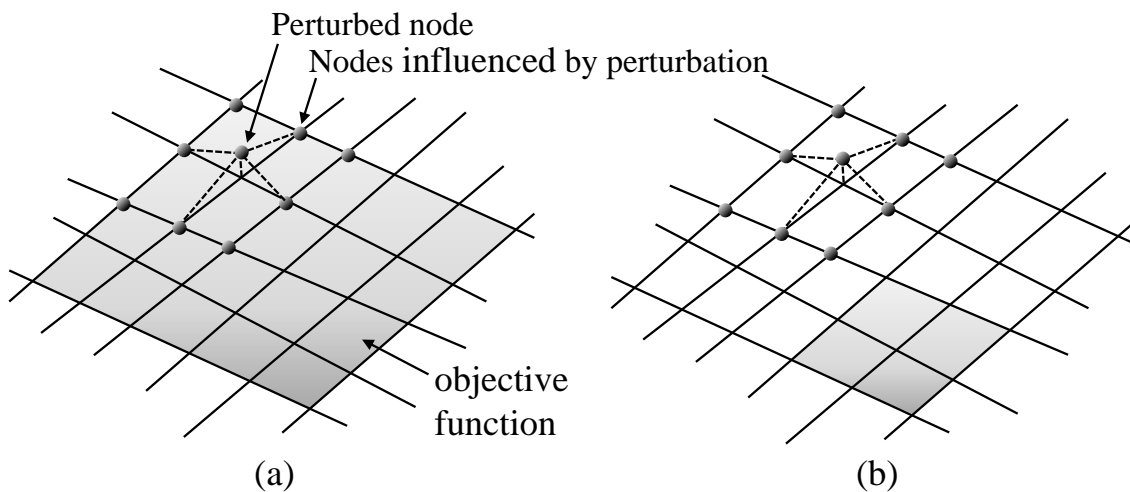


Figure 3.7: Optimization workflow with respect to the different modules and tools

will be demonstrated with the example in section 4.3.

Furthermore, much effort has been spent to consider the sparsity of the matrices and vectors. For example, in the computation of the gradients a reduced effort is achieved by screening the partial derivatives  $\partial J/\partial \mathbf{s}$  and  $\partial J/\partial \mathbf{u}$  of equation (2.5) for zero entries. Indeed for the stress objective function a perturbation  $\Delta \mathbf{s}$  or  $\Delta \mathbf{u}$  at a certain node only changes the stresses at the node itself and at the nodes on the neighboring elements, see figure 3.8(a). According to figure 3.8(b) in case none of the nodes is in the set of the objective function, the function value in fact does not change and consequently the entry in the gradient vector for the perturbed node is zero. Whether a vector entry is zero depends on the mesh topology and on the set of the objective function. Thus, this information is valid in every optimization iteration.



**Figure 3.8:** Impact of perturbation of a node on the objective function; (a) perturbed and neighboring nodes are in the set of the objective function (grey area) and (b) perturbed and neighboring nodes are not included in the set

---

# Application of Vertex Morphing

---

In the previous chapters the theory for node-based shape optimization was derived. In detail, chapter 2 provides the theoretical background for the adjoint sensitivity analysis, its transfer to specific response functions, the consistent out-of-surface regularization of the gradient field and the regularization of the volume mesh as well as the mesh update. Chapter 3 concentrates on the complete parameter free shape optimization workflow and the aggregation of constraints. The whole process is designed to optimize complex three-dimensional industrial applications robustly and efficiently.

At first, the success of this project is studied on three academic examples in 2D and 3D where the results can easily be compared with the analytic solution or the work of others. After that two large geometrically complex parts from aerospace industry are investigated.

In order to avoid repeating explanations about the filter, in all the applications the Gaussian filter from equation (2.47) is used. The filter radius is defined in terms of the standard deviation  $\sigma$  and the Gaussian function is set to zero in a distance of  $3\sigma$  from the center. To guarantee the unit-integration-condition of the filter, the post-scaling approach [Ble13] is proposed. Moreover, an additional weighting with respect to the direction of the design variables is implemented by decreasing the influence of a design variable onto the filter if the angle between

the considered design variable and the neighboring design variable increases. This is achieved by adding the scalar product of the considered design variable direction and the direction of the neighboring design variable as additional factor in the computation of the filter matrix  $\mathbf{A}$ . Per definition, up to  $90^\circ$  an interaction is supposed. If the angle exceeds this value they do not influence each other and the scalar product is set to zero.

## 4.1 Holes in a plate

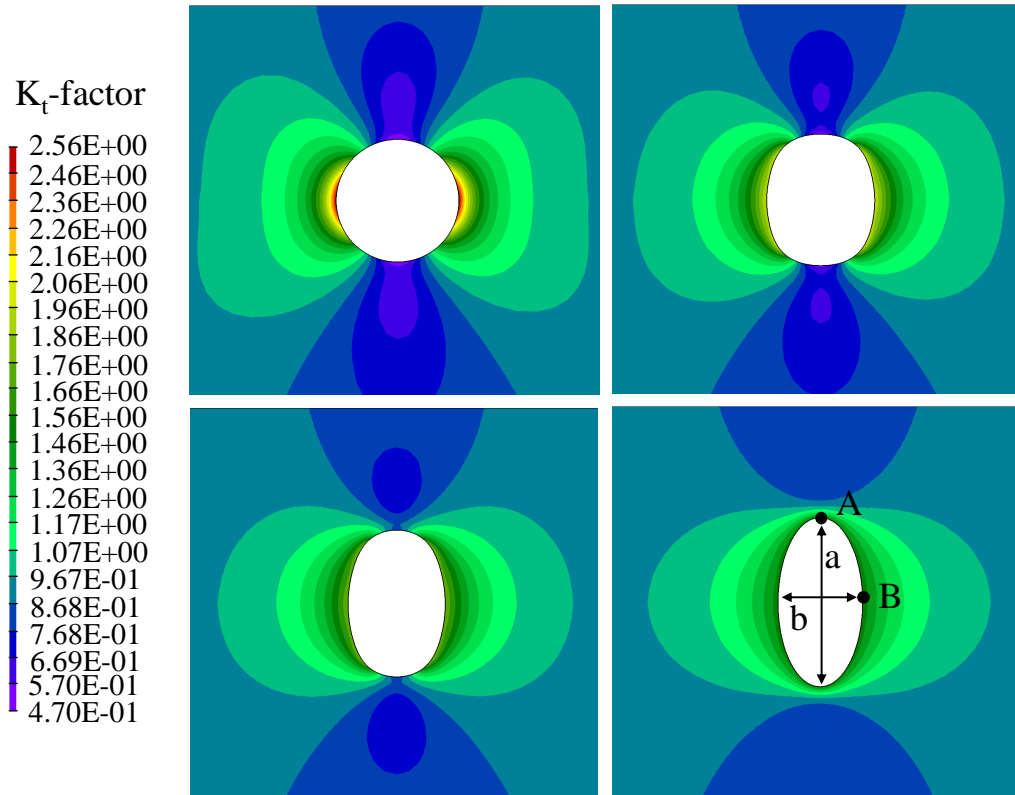
Reduction of notch stresses in aircraft structures is a common task since these are often fatigue-prone locations. As an example cutouts in sheet metals for air ducts or bolt holes can be mentioned. For manufacturing reasons they are in many cases circular in shape but ideally the optimal hole geometry has to be determined to minimize the stress concentration. As this shape optimization problem has a major benefit on the fatigue life of the part it has been subject of many studies.

The presented two-dimensional problems are modeled with one row of linear elements and the thickness in the third dimension is chosen to be larger than the span of the filter function to avoid any interaction between the nodes on both layers. Consequently, the setup is perfectly equivalent to a 2D simulation as the model remains unchanged in the third dimension until the end of the optimization. Moreover, in both models the distance between the holes and the borders is defined large enough that no remarkable interaction takes place.

### 4.1.1 Single hole in a plate

The first example considered here is a quadratic plate with a single circular hole subjected to biaxial in-plane stresses  $\sigma_1$  and  $\sigma_2$  according to figure 2.16. This problem has already been introduced in the context of regularization of the raw sensitivities in section 2.6 and is now evaluated with respect to the optimal design. As an additional information the filter radius has been defined as 15 % of the hole diameter and the aggregation parameters of the KS-function are defined

as  $\rho = 15$  and  $\bar{\sigma} = 1.5\sigma_1$ , respectively. For the sake of comparison with the analytical solution, the resulting stresses have been normalized with  $\sigma_1$  which yields the  $K_t$ -factor.



**Figure 4.1:** Evolution of the design and the  $K_t$ -factor from the initial design, at top-left, to top-right, bottom-left and bottom-right

According to Peterson [Pet74] the worst stress concentration factor for the circular hole is analytically derived as

$$K_t = \frac{\sigma_{max}}{\sigma_1} = 3 - \alpha, \quad (4.1)$$

where  $\alpha$  is the ratio between  $\sigma_2$  and  $\sigma_1$  being 0.5 in this example. Thus, the corresponding  $K_t$ -factor for the initial design is computed to 2.50 which fits very well with the result of the Finite Element analysis where the  $K_t$ -factor is 2.56, cf. figure 4.1 at top-left. Since only the mass constraint is applied which is identical to a constraint that the area of the hole must not decrease, the optimal solution is well known. The circular hole transforms to an elliptical one with the ration of its axis of 2/1 which exactly reflects the ratio of the applied stresses  $\sigma_1$  and

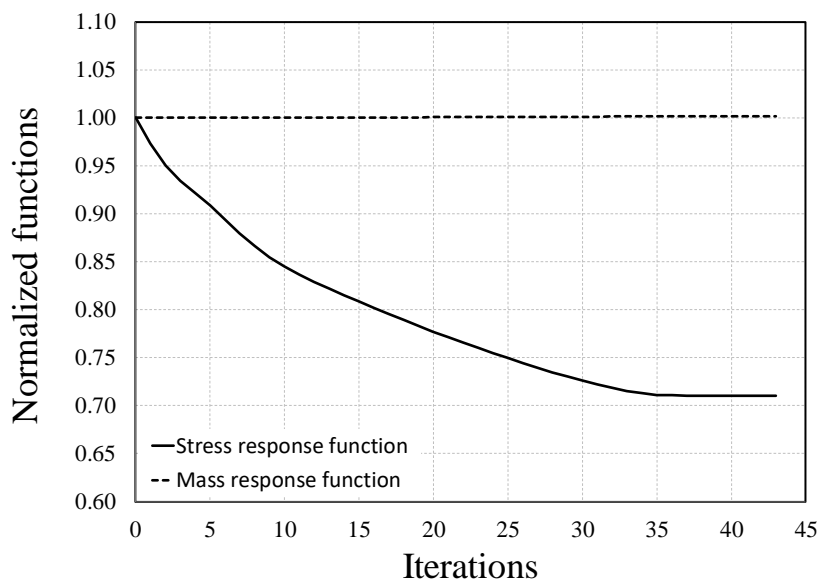
$\sigma_2$ . The analytically derived  $K_t$ -factor in [Pet74] for the elliptical hole can be computed for the positions  $A$  and  $B$  in figure 4.1 at bottom-right as follows,

$$\begin{aligned} K_{t,A} &= \left(1 + \frac{2a}{b}\right) \frac{\sigma_2}{\sigma_1} - 1, \\ K_{t,B} &= \left(1 + \frac{2b}{a}\right) - \frac{\sigma_2}{\sigma_1}. \end{aligned} \quad (4.2)$$

Taking the optimal geometry where  $a = 2b$ ,  $a$  and  $b$  are the major and minor axis of the ellipse, the boundary condition  $\sigma_1 = 2\sigma_2$  and inserting it in equation (4.2) yields

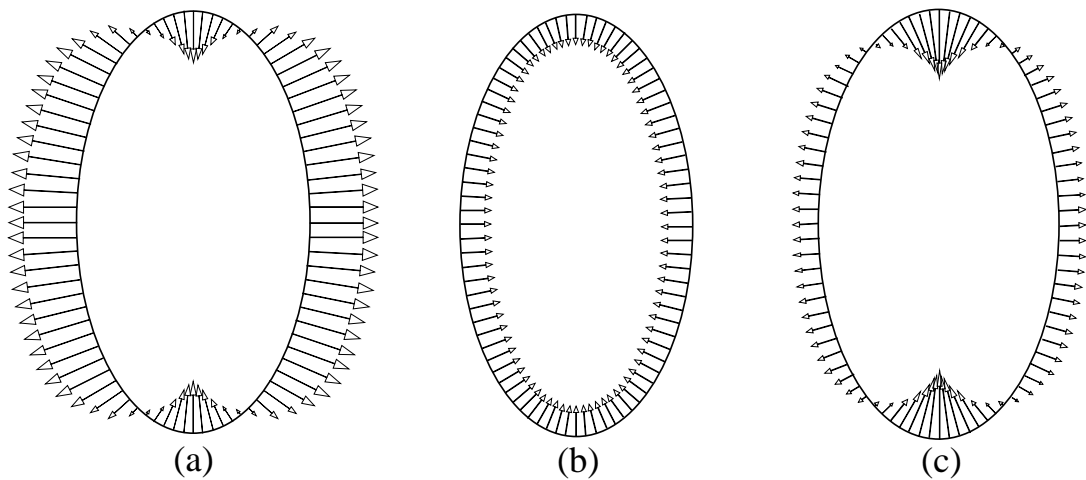
$$\begin{aligned} K_{t,A} &= \left(1 + \frac{2a}{b}\right) \frac{\sigma_2}{\sigma_1} - 1 = \left(1 + \frac{2 \cdot 2b}{b}\right) \frac{\sigma_2}{2\sigma_2} - 1 = 1.50, \\ K_{t,B} &= \left(1 + \frac{2b}{a}\right) - \frac{\sigma_2}{\sigma_1} = \left(1 + \frac{2b}{2b}\right) - \frac{\sigma_2}{2\sigma_2} = 1.50. \end{aligned} \quad (4.3)$$

It is interesting to notice that for the optimal geometry  $K_{t,A} = K_{t,B} = 1.50$  which indicates that the stress concentration is completely homogeneous around the elliptical hole. This optimal solution can also be seen in figure 4.1 at bottom-right where the final design of the shape optimization is illustrated. With a constant  $K_t$ -factor of 1.53 around the hole, the value approaches almost the analytical solution after 43 iterations, cf. figure 4.2.



**Figure 4.2:** History for the normalized stress and mass response function





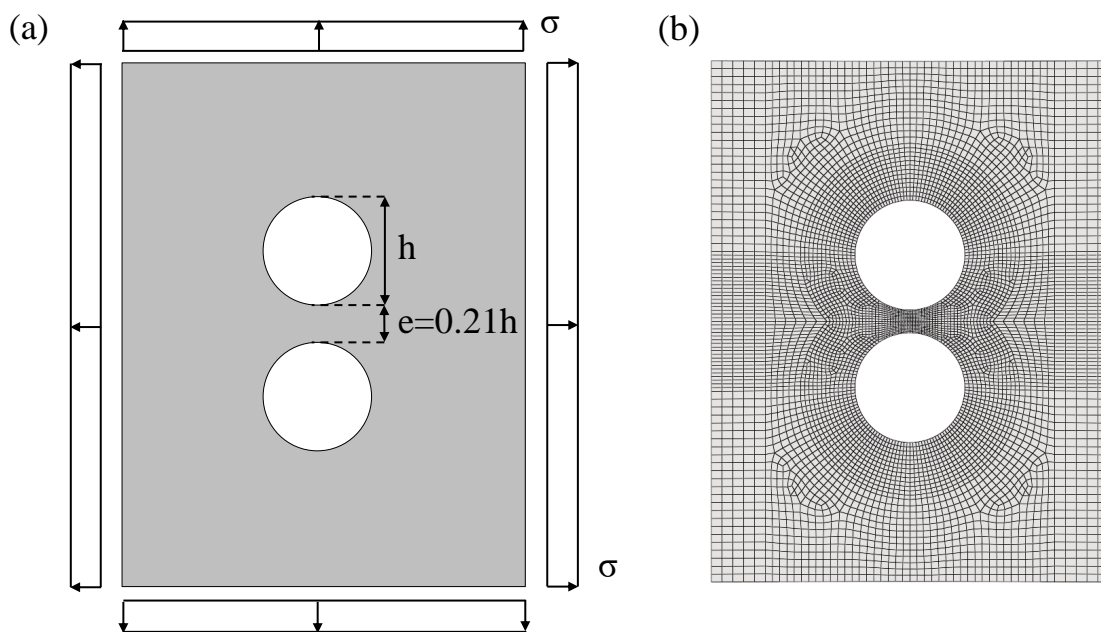
**Figure 4.3:** Gradients of the stress response function (a) and mass response function (b) as well as projected gradient (c) at iteration 30

An essential process step necessary to find the constrained optimum at all is the aggregation of constraints in the descent direction. As described in section 3.2.2, Rosen's gradient projection method is applied projecting the gradient of the response function on the subspace tangent of all active constraints. The functionality of the projection can descriptively be demonstrated on the basis of this example. Figure 4.3(a) shows the unconstrained shape update at optimization iteration 30 for the stress response function reducing the stresses in direction of the steepest descent. In figure 4.3(b) the gradients of the mass response function are shown always leading to a constant offset of the design variables as discussed in section 2.5. In this special case of a uniform gradient vector of the constraint function, the projected gradient vector is still similar in shape compared to the gradient vector of the objective function. It is simply shifted such that the constraint is fulfilled, see figure 4.3(c). In case of a nonuniform gradient vector or more than one active constraint the projected gradient vector can have a completely different shape. However, taking the projected gradient as descent direction in every optimization iteration prevents the violation of the constraints. In figure 4.2 the history of the normalized mass is evaluated confirming that during the whole optimization the weight is not increased.

Though, the optimal solution for the minimization of the notch stresses is trivial, it is an illustrative example to demonstrate the potential of the proposed process.

### 4.1.2 Two closely-spaced holes in a plate

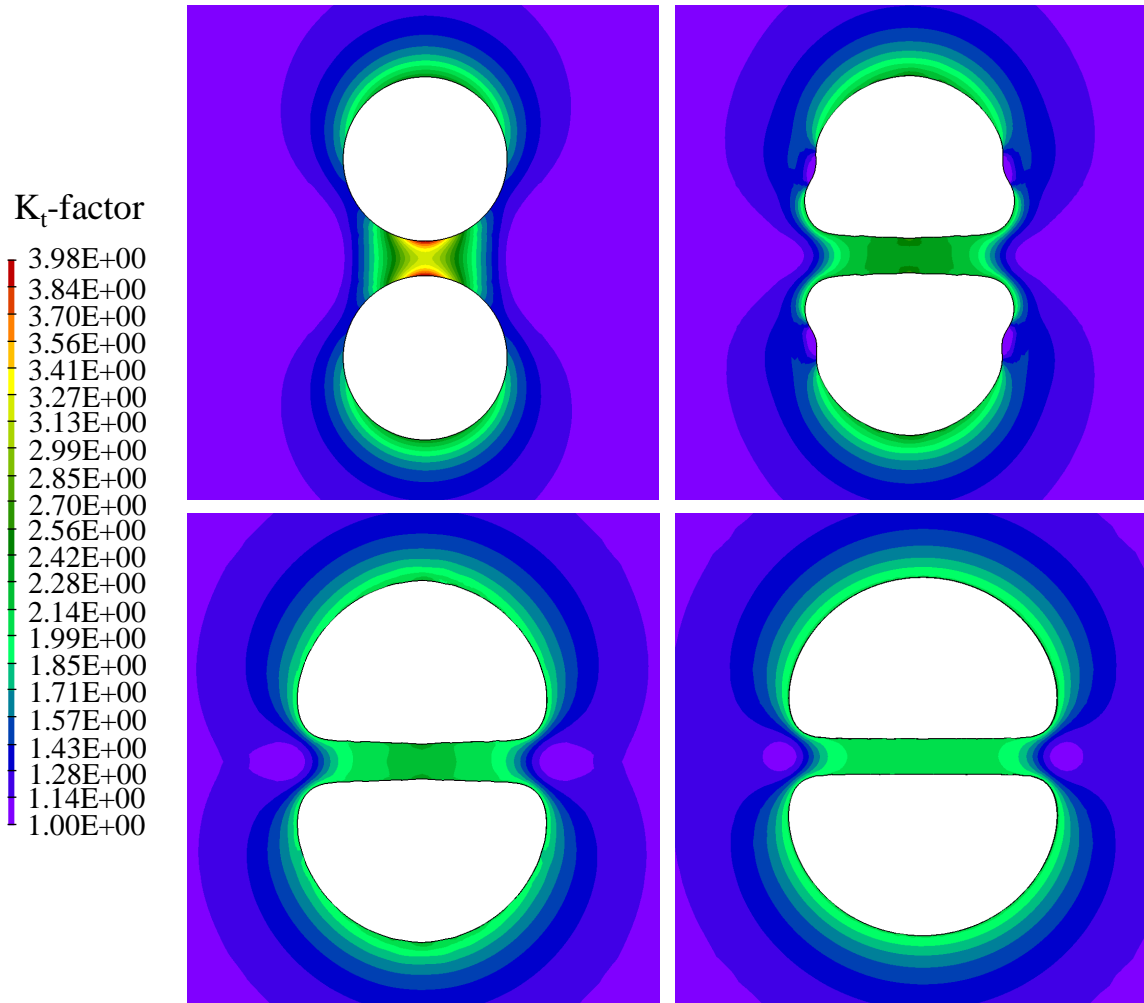
The second conceptual example is a two-dimensional biaxially loaded plate with two closely-spaced holes illustrated in figure 4.4(a). As aforementioned, in the past much effort has been spent to reduce the stress concentration by finding the free-form shapes of holes and fillets for various problems. Among many others, Waldmann et al. [WHR03] analyzed the stress concentration of the underlying mechanical model and optimized the shape of the holes in the context of reworking where only material removal is allowed.



**Figure 4.4:** Plate with two closely-spaced holes, mechanical model (a) and Finite Element discretization (b)

Figure 4.4(b) shows the discretized Finite Element model for the shape optimization which consists of linear elements and the design variables are defined on the boundaries of the two holes which also serve as set for the evaluation of the response function. Regarding the main dimensions, the diameter of both holes is defined as  $h$  and the distance  $e$  between both holes is  $0.21h$ . The plate is uniformly loaded in all directions by a traction force  $\sigma$ . The optimization task can be formulated as a stress minimization problem at the boundaries of the two holes while all movements of the design variables are constrained to

their negative normal direction such that only a displacement in outward direction is permitted. This constraint considers the reworking since the size of the holes can only be increased due to the repair. As described in section 2.3 the aggregation parameter  $\rho$  determines how close the KS-function approximates the largest stress value. Unlike in real applications where the focus is on sufficiently

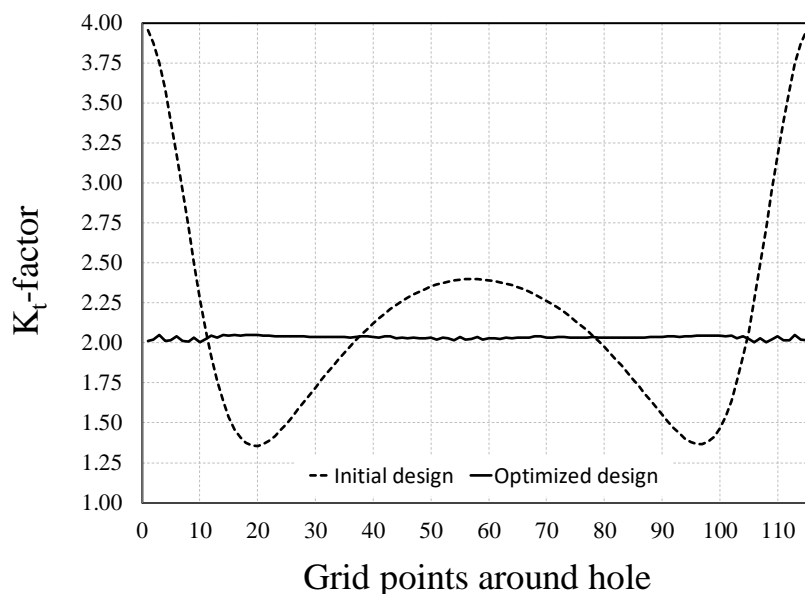


**Figure 4.5:** Evolution of the design and the  $K_t$ -factor from the initial design, at top-left, to top-right, bottom-left and bottom-right

improving the response function regardless of the value of  $\rho$ , the goal in this example is to obtain the optimum solution from [WHR03]. For such a case,  $\rho$  has to be defined very high which leads to numeric problems [AFB10] or alternatively increased in the course of the optimization. Here, the latter option is chosen and  $\rho$  is increased in the course of the optimization from 10 to 40. The second aggregation parameter  $\bar{\sigma}$  is held constant with a value of  $2\sigma$  and the

initial step size  $\alpha_0 = 0.001h$  is defined very small to exactly trace the nonlinear displacement of the design variables. The standard deviation of the Gaussian kernel for the out-of-surface regularization is assumed to be  $0.075h$ . Generally, smaller filters allow for a faster approach to the optimal geometry, especially if the optimum includes high curvatures. Nevertheless, the filter can not be smaller than a certain limit due to stability reasons thus the standard deviation of the filter is only reduced in the last iterations to  $0.025h$  when the shape nearly converged to the optimal design.

According to [WHR03], the  $K_t$ -factor for the aforementioned circular holes is 3.96 at the worst position occurring in the symmetry plane between them. After the optimization the notch stresses are reduced to a factor of 2.02. Key element of the obtained geometry is that the interaction effects between the holes are completely eliminated. It is also interesting to note that the  $K_t$ -factor for two interacting optimal holes is identical to that for the corresponding optimal shape for a single hole presuming identical boundary conditions.

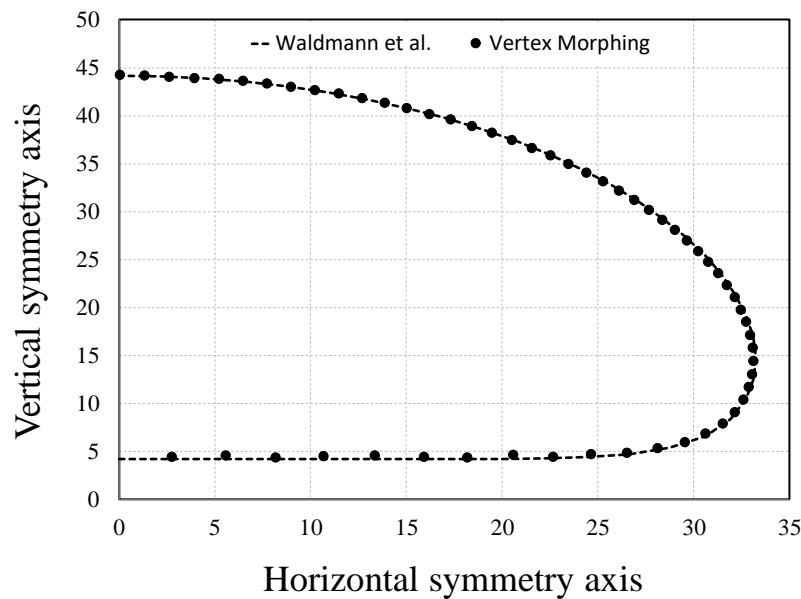


**Figure 4.6:** Distribution of  $K_t$ -factor around the holes evaluated at the grid points

Regarding the results obtained with the Vertex Morphing method, the  $K_t$ -factor of the initial geometry fits with 3.98 very well to the results of [WHR03]. Furthermore, figure 4.5 illustrates that the notch stresses are continuously reduced

in every iteration until the final design is reached (figure 4.5 at bottom-right) where the  $K_t$ -factor approaches 2.06 only deviating by 2 % from the results of [WHR03]. In figure 4.6 it can be seen that the interaction between both holes is also eliminated and the  $K_t$ -factor is distributed evenly around the hole. Finally, figure 4.7 verifies that the optimal shape computed with the Vertex Morphing method and the one from [WHR03] fit very well.

Beside the fact that the optimal solution can successfully be obtained, the proper treatment of mesh is also worth to point out in this example. The introduction of large geometry changes and especially the reduction of curvature is a very challenging task in shape optimization. To get a rough idea about the magnitude of the shape change, the nodal displacement in the course of the optimization is approximately twelve times larger than the length of the corresponding element edge. In such cases the optimization often fails due to the deterioration of the mesh quality or the process diverges since no meaningful shape can be found by the variables further reducing the objective function. As seen in figure 4.5 the circular holes converge smoothly to the optimal design without any mesh irregularity.

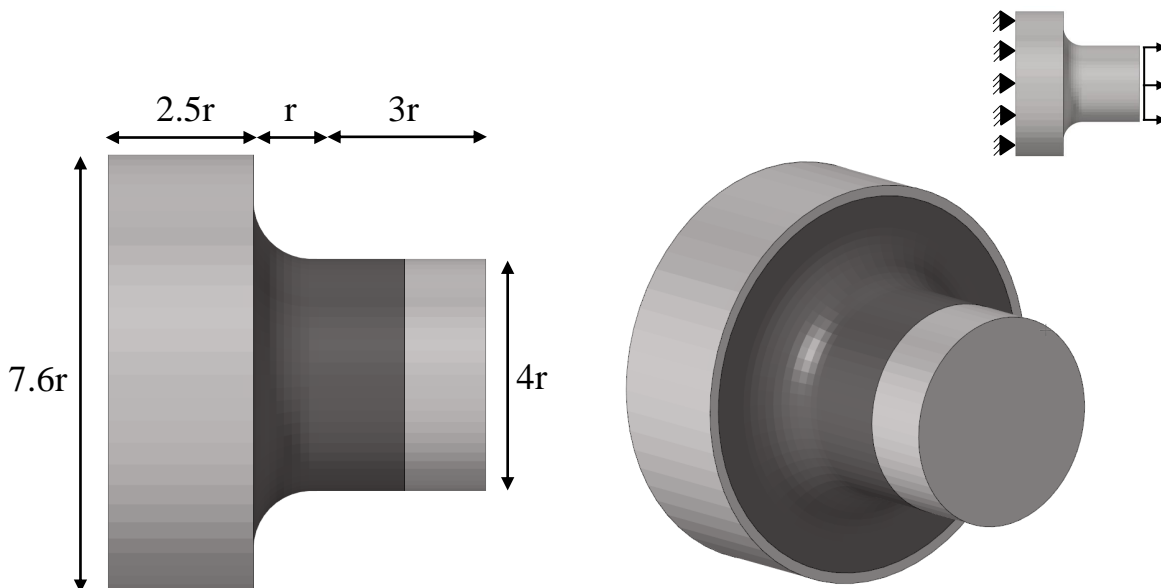


**Figure 4.7:** Comparison of the optimized shapes

## 4.2 Fillet design

In this section the shape optimization of a fillet at the transition of two cylinders under tension is studied. More precisely, the purpose of this example is to discuss two important aspects. The first one is the extension of the shape optimization process to three-dimensional problems and to the commonly used quadratic element formulations. The second aspect is the influence of two important parameters of the optimization which are the filter radius and the step size of the optimization algorithm.

The geometry and the boundaries of the fillet design are axially symmetric (figure 4.8). The fillet is defined with a radius  $r$  and merges tangentially in the two cylinders. The bigger one has a diameter of  $7.6r$  with a thickness of  $2.5r$  and the smaller one has a diameter of  $4r$  being  $3r$  thick. The face of the bigger cylinder is constrained in all degrees of freedom. The shape optimization is performed for a tension load acting on the front face of the smaller cylinder. The design surface is depicted in a dark grey color and consists of 4140 design variables.

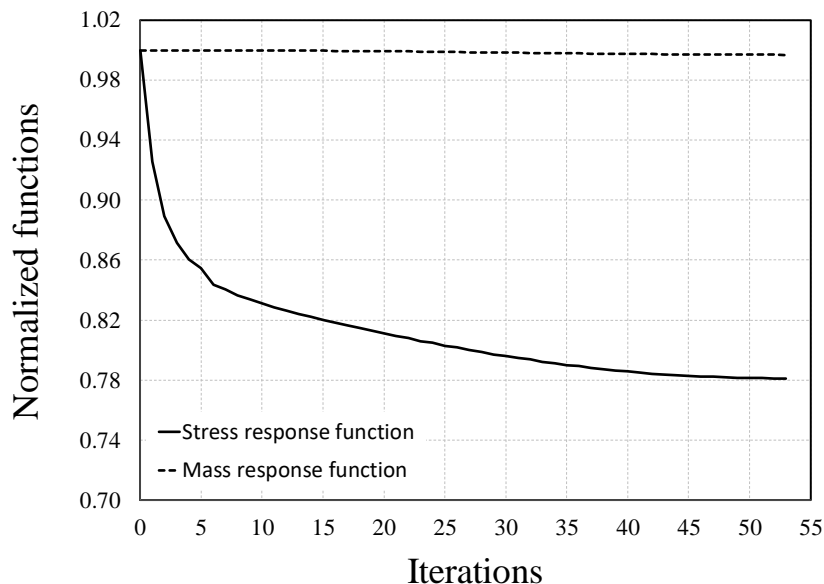


**Figure 4.8:** Initial geometry of fillet design. The design surface is marked in dark grey. At the top-right, the mechanical system with the boundary conditions is shown.

### 4.2.1 Reduction of notch stresses

The goal of the optimization is to minimize the stresses in the transition between both cylinders under the constraint of at least preserving the initial mass. Regarding the chosen filter, the standard deviation of the Gaussian kernel is defined to be  $0.18r$  which spans approximately 8 elements. The initial step size is about 13 % of the element size.

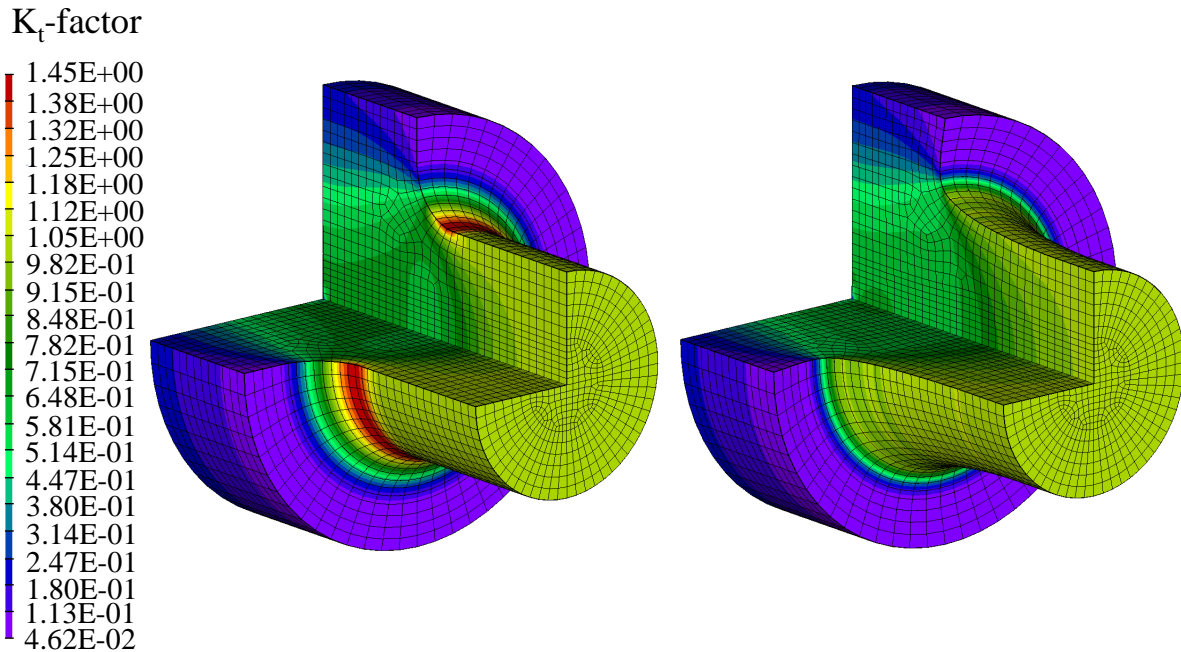
Figure 4.9 shows the optimization history of the stress objective function which converged after 53 iterations to a local minimum and achieved about 22 % of improvement. After that the sensitivities are very small and the objective function remains almost unchanged. Regarding the mass, the constraint is active during the whole optimization and thus is retained by the Gradient Projection method almost at its initial value. The initial and the optimized design are compared in figure 4.10. In the course of the optimization the stress concentration factor  $K_t$  decreased from 1.45 to 1.06 which is an improvement of 27 %. The reduction



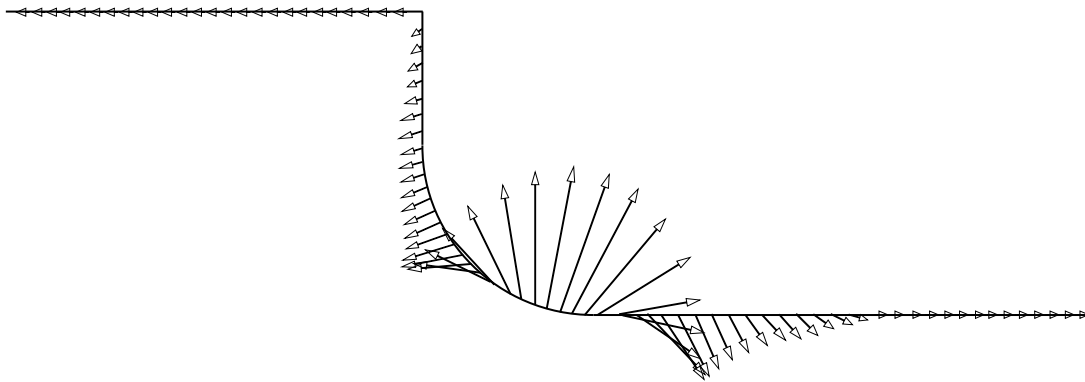
**Figure 4.9:** Optimization history of the normalized response functions

of the stresses is mainly achieved by pulling the wall boundary in the region of the highest stresses out and pushing it in on both sides next to this area. In the case of a curved boundary this is equally to flattening the curvature. The corresponding shape update vectors at the first iteration can be seen in figure

4.11. Vertex Morphing transforms the sudden change of the surface sensitivity to a meaningful geometry. Comparing the design with CAD, this would not be feasible with a standard parametrization.



**Figure 4.10:** Evolution of  $K_t$ -factor from the initial design at the left to the optimized one at the right.



**Figure 4.11:** Combined in- and out-of-surface update vectors at the longitudinal upper design boundary in the first iteration.

Furthermore, this example also reveals the advantage of the improved Traction method (section 2.6.5). Concentrating on the update vectors of the invariable areas, they also contribute to the shape update due to their displacement in tangential to the surface direction. Generally, the in-plane component of the update



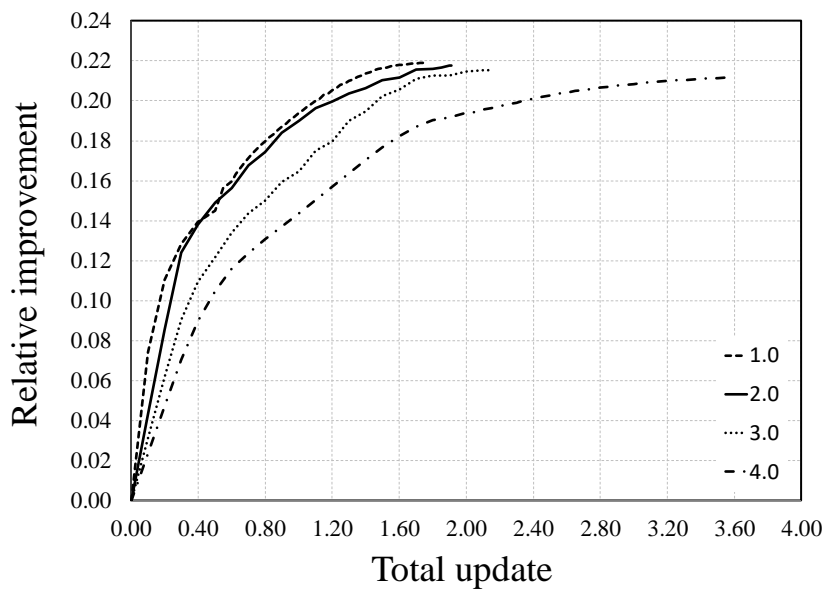
vectors computed by the Traction method supports to reduce the mesh distortion.

Additionally, as it can be seen in figure 4.10, the regularization of the quadratic elements works very well and results in smooth shapes throughout the optimization.

### 4.2.2 A filter and step size study

In this section the influence of the step size and the filter radius on the basis of the presented fillet optimization example is reviewed.

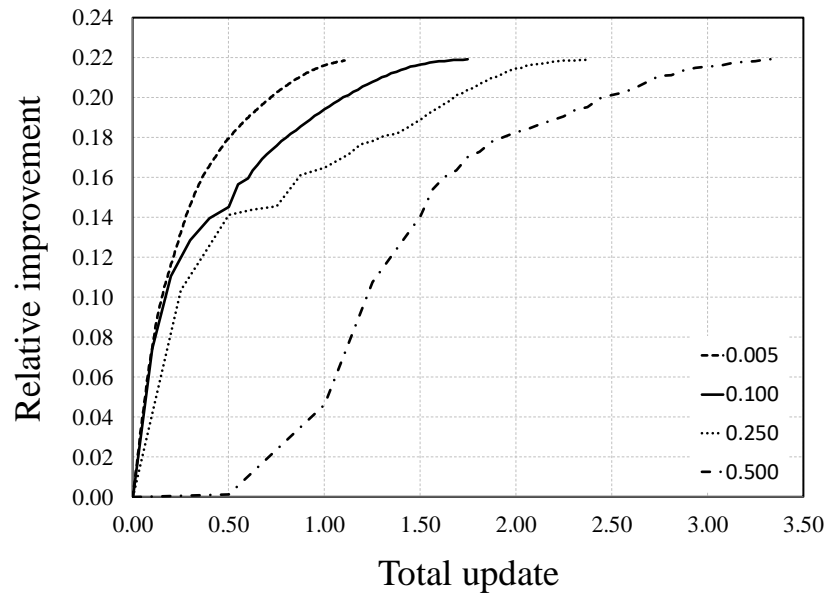
As discussed in the sections 2.6.2 and 2.6.3, the filter radius is defined as the standard deviation of the Gaussian kernel used to create the filter matrix  $\mathbf{A}$  of the Vertex Morphing method. Demonstrated by [HSB14] and also concluded on the basis of this example, the smaller the filter radius the faster the improvement of the objective function. Figure 4.12 compares the objective improvement for four different filter radii as a function of the total design update defined as the sum over the step sizes in every iteration. This measure has to be chosen as due to the Backtracking algorithm the step size in an iteration may be different for different filter radii though the initial step size is equal. The final geometry in all four cases is similar with small differences in the curvature. This leads to slightly higher  $K_f$  values for higher filter radii which explains the marginally worse improvement of the objective. As already mentioned, the rate of improvement of the objective function is the highest for the smallest filter radius. Of course there are certain limitations on the size of the filter radius. Having in mind that the filter is needed to regularize the ill-posed shape optimization problem it can not be defined arbitrarily small. There is a certain limit where the regularization intensity of the  $\mathbf{A}$  operator is too small so that the stability criterion is not fulfilled and shape irregularities rapidly occur. Therefore, the filter function has to include a sufficient number of grid points. A practical approach to avoid the definition of the filter size as a fixed distance, is to define it in dependency of the surface element size. This guarantees a minimum number of grid points evaluated in the filter function. Beside the numerical stability there are also some practical



**Figure 4.12:** A filter radius study for the fillet geometry. The total update is defined as the sum over the step sizes in every iteration.

reasons requiring a sufficiently high filter radius. Due to production limits and costs a certain level of smoothness as well as a minimum curvature with less geometrical details is desired.

The shape optimization problem is in general nonlinear. As an approximation the gradient information is linearized and multiplied with a finite step size in every optimization iteration. Therefore, in the following step size study the influence of the step size of the Steepest Descent method on the objective function is studied. Theoretically, with an increasing step size the deterioration of the optimizer from the real direction to the local optimum also increases. As demonstrated in figure 4.13 on the basis of various step sizes this consideration applies also in practical applications. The smaller the step size the faster is the rate of improvement of the objective function. However, the final design is not affected by the chosen step size since the geometry is identical in all four cases with an improvement of 22 %. This is achieved by the Backtracking algorithm which bisects the step size until the local minimum is reached. Regarding the computational cost, very small step sizes require many iterations until the design has converged whereas larger steps proceed faster taking several cutbacks into account.



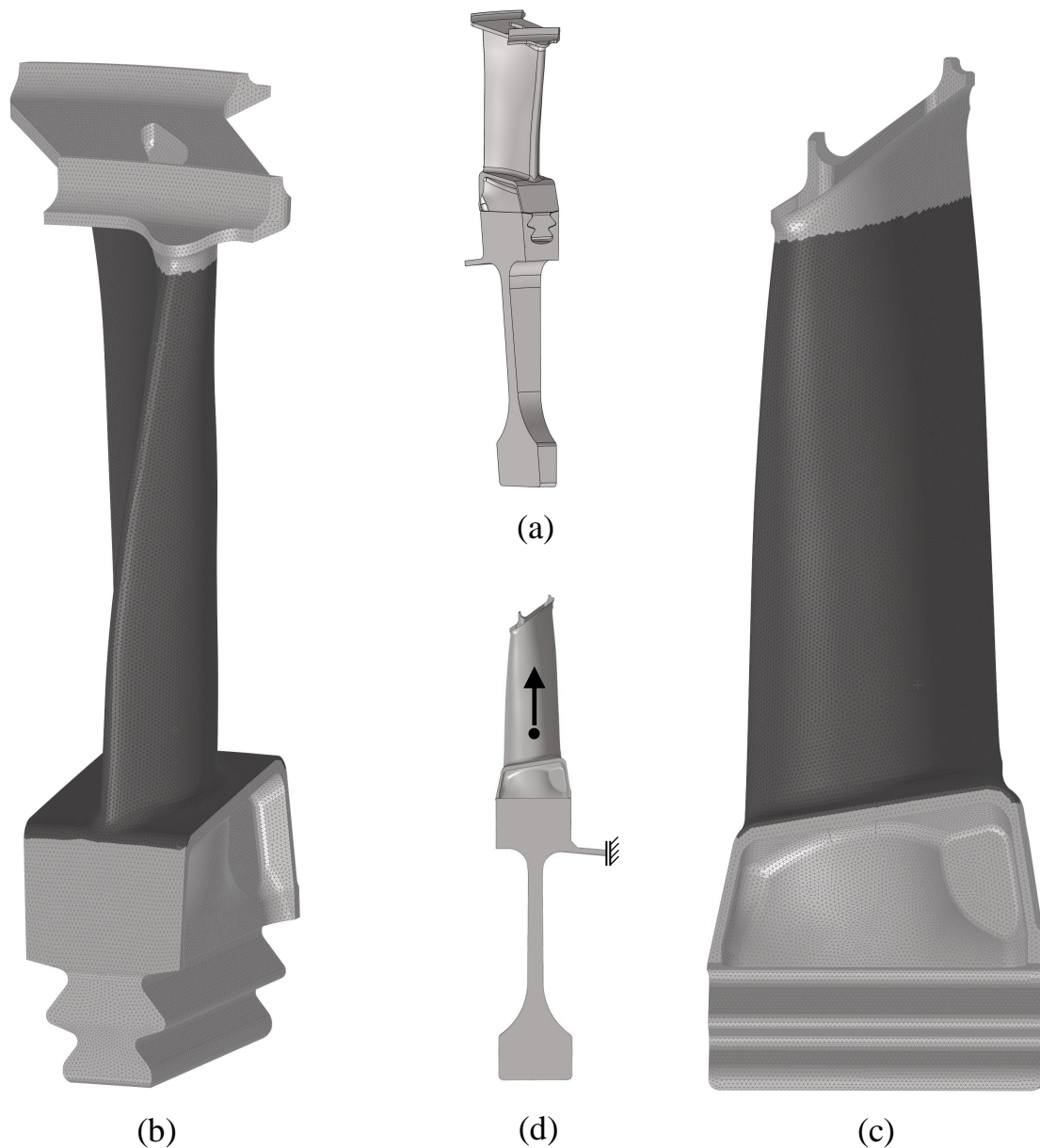
**Figure 4.13:** A step size study for the fillet geometry. The total update is defined as the sum over the step sizes in every iteration.

### 4.3 Low pressure turbine blade

In this section the Vertex Morphing method is applied to the shape optimization problem of a low pressure turbine blade. Compared to the academic examples this problem is much larger in computational size with a highly complex three-dimensional geometry. First, the constraint shape optimization is performed. Then, the capability of the proposed optimization process is discussed on the basis of a time consumption study.

#### 4.3.1 Constrained optimization of stresses

In this section the shape optimization of the blade shown in figure 4.14 is performed. The model consists of 725.407 quadratic tetrahedral elements and 1.114.091 nodes resulting in an average element edge length of 0.5 mm. The design surface is highlighted in dark grey in the figures 4.14(b) and 4.14(c) including 93.330 nodes. The objective function is the reduction of the stresses according to equation (2.25) and also evaluated in the grey area. The corresponding aggregation parameters  $\bar{\sigma}$  and  $\rho$  are set to 300 and 10, respectively. An important aspect in the optimization of aerodynamic structures is the limitation

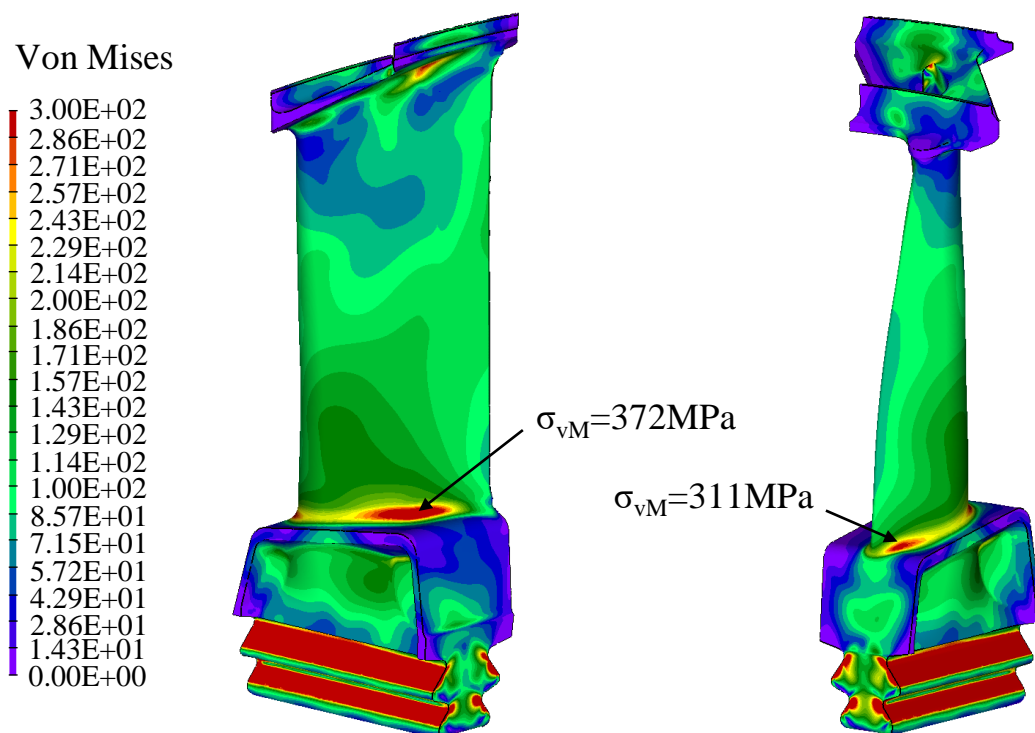


**Figure 4.14:** Cyclic symmetric model of a bladed rotor (a), the discretized blade with the design space marked in dark grey (b)(c) and the mechanical system with the bearing conditions and the centrifugal load (d).

of the design change and the conservation of the main design features to restrain potential negative effects on the aerodynamic behavior of the component. To this end, large enough filters are used such that the improvement pattern is smooth and the overall shape is not disturbed. The chosen filter radius approximately spans 18 elements which corresponds to a standard deviation of 3 mm of the Gaussian function. Furthermore, a displacement constraint of 1.0 mm for every design variable is defined assuming that up to this value no effect on the aerody-

namics takes place. In addition, as the weight is always an important quantity in aerospace industry it is not allowed that the optimized design exceeds its initial value.

The rotation of the rotor system (figure 4.14(a)) exerts a centrifugal load on the blade which causes high stresses particularly in the transition of the airfoil and the blade root. Moreover, a temperature gradient induced by the hot flow in the gas path causes additional thermal stresses which have to be superimposed. The resulting von Mises stresses are shown in figure 4.15 where the life limiting positions occur in the aforementioned transition zone with 311 MPa at the leading edge and 372 MPa at the trailing edge.



**Figure 4.15:** Von Mises stresses for the initial design caused by mechanical and thermal loads.

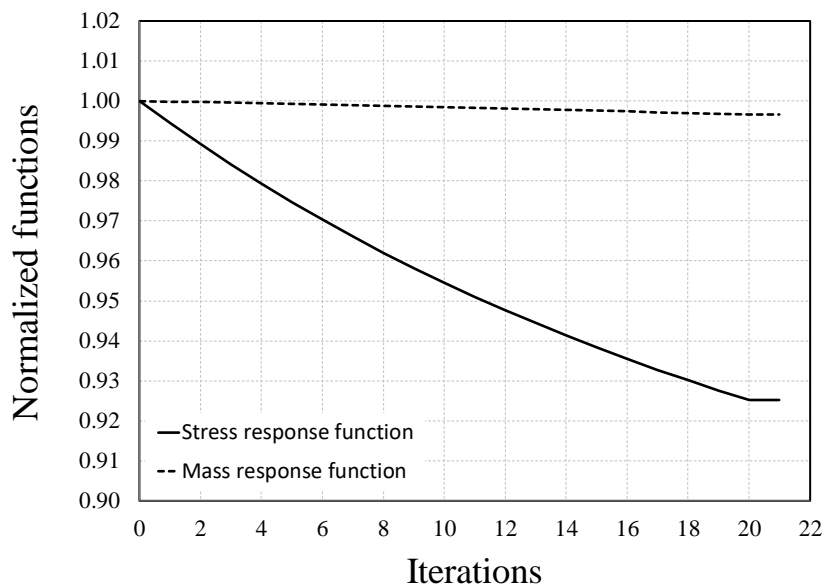
In what follows, the section is divided in two parts: First, some brief remarks on the computational loop are discussed. Then, the improved design is presented and compared with the initial one.

The optimization loop starts with the static analysis including the mechanical

and thermal loads based on the discretized model shown in figure 4.14. As a result, the global stiffness matrix, the external load vector as well as the stress and displacement fields are obtained. Based on these quantities the gradients of the objective function and the constraints are evaluated from equation (2.16) and repeated here for clarity

$$\frac{dJ}{ds} = \frac{\partial J}{\partial s} + \lambda^T \cdot \left( \frac{\partial \mathbf{K}}{\partial s} \cdot \mathbf{u} - \frac{\partial \mathbf{f}}{\partial s} \right). \quad (4.4)$$

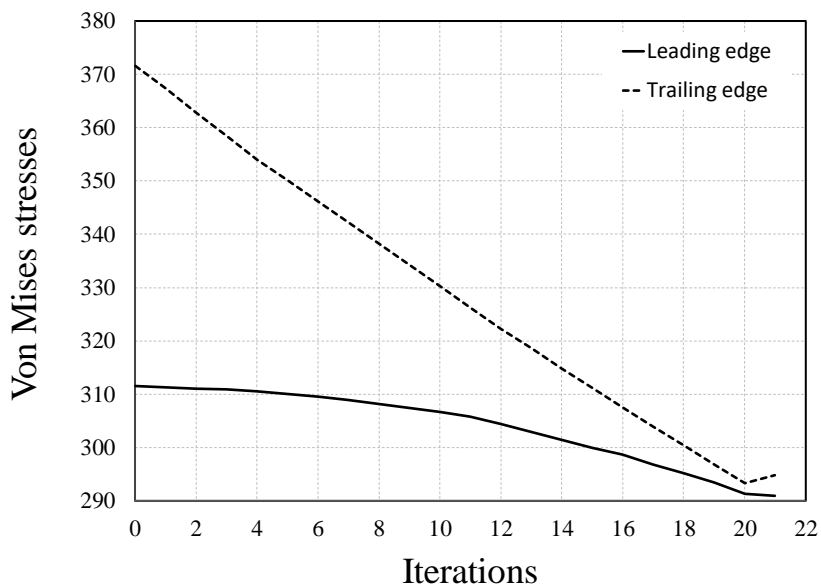
The Gradient Projection method uses these sensitivity fields to determine the shape update vector which lies on the subspace tangent of all active constraints. In the sequence, the improved Traction method (section 2.6.5) is applied to update the surface mesh with the shape update vector as prescribed displacement. Herein, the distance dependent Young's modulus is distributed within the adjacent two elements rows. Having the updated nodal coordinates for the whole mesh the next iteration is started with the computation of the physical problem. The loop is continued until the optimization problem has converged.



**Figure 4.16:** History for the normalized stress and mass response functions.

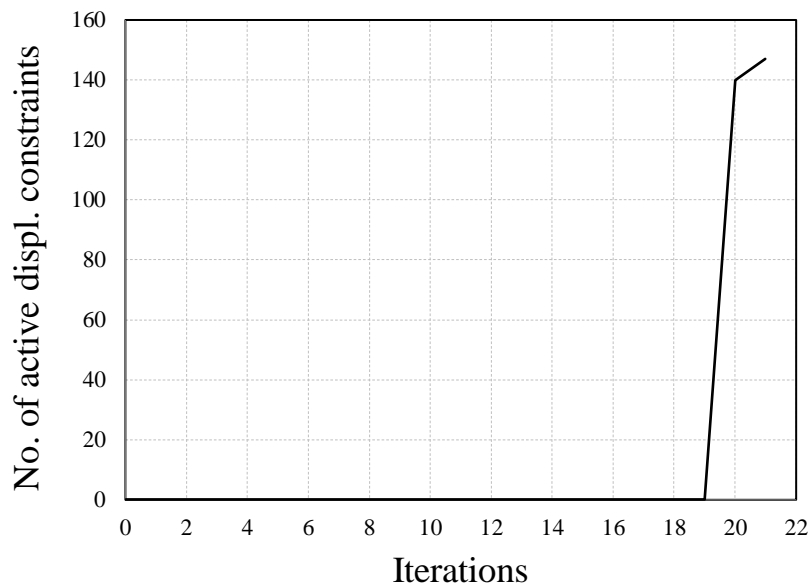
Figure 4.16 shows the history of the optimization for the stress response function. All in all the Vertex Morphing method needed 21 iterations to improve the design remarkably until convergence. Around 7.5 % improvement in the KS-

function which corresponds to a reduction of 6.4 % (291 MPa) at the leading edge and 21 % (295 MPa) at the trailing edge, respectively, can be achieved. In aerospace industry this is a huge improvement as the designs typically challenge the limits with several optimization tasks being involved. For instance, the airfoil is optimized with respect to its aerodynamic properties and its dynamic behavior. Comparing the stress history for both positions in figure 4.17 in detail reveals the advantage of the KS-formulation. In the first optimization iterations mainly the trailing edge position showing the highest stresses is improved while the leading edge stresses are more or less retained at the same level by the KS-function. For example, the maximum stress function from equation (2.24) would not track the leading edge stresses at all even if they would increase due to the fact that only the worst position is taken into account. As the stresses are more and more aligned in the course of the optimization the influence of the leading edge position on the KS-function increases leading to a steeper slope of stress reduction. At the end of the optimization both positions nearly have the same degree of capacity utilization. This property of the KS-function is an important advantage for the development of equally loaded and robust designs.



**Figure 4.17:** History for the von Mises stresses at the leading and trailing edge.

Regarding the behavior of the constraints during the optimization, figure 4.16 shows the history of the mass constraint and figure 4.18 tracks the number of



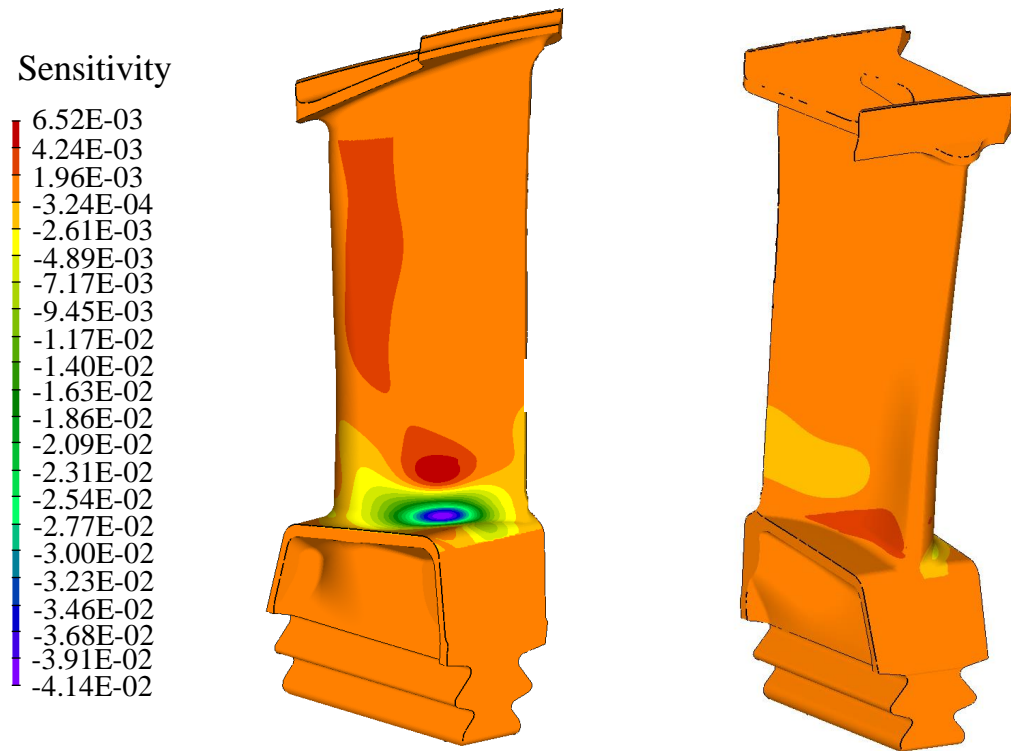
**Figure 4.18:** History for the number of active displacements constraints.

active geometric constraints in every iteration. In every iteration the normalized mass is successfully kept below the value of 1.0 by the Gradient Projection method. Thus, the design in every iteration can be considered feasible with respect to its weight. As aforementioned, the geometrical constraint was defined to become active at a maximum displacement of a design variable of 1.0 mm. Up to iteration 19 all variables are still below that limit and the number of active geometric constraints is consequently zero. At iteration 20 the constraint becomes active at 140 variables in the region of the trailing edge leading to a kink in the objective function as suddenly the design can not be improved anymore in this region. Though the optimization algorithm found once more a feasible design, the objective function could not be further reduced and the optimization finished.

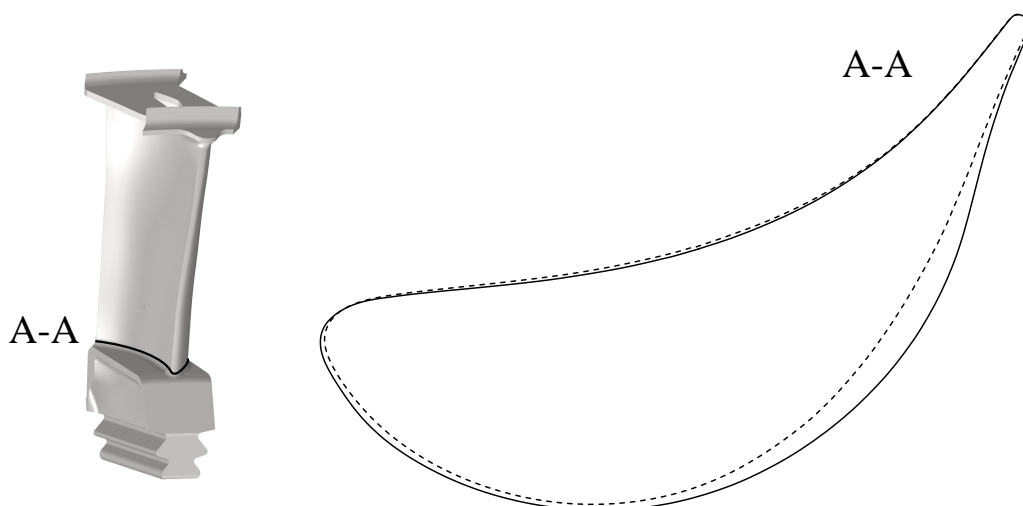
The improvement of the design can also be seen through the sensitivity map on the blade. Figure 4.19 displays the surface sensitivity of the stress objective function of the initial design. The regions with the highest sensitivity undergo the largest deformation. This is also confirmed in the figures 4.20 and 4.21 where the initial and the final shape at two different cross sections is compared. During the optimization the cross-section is thickened up mainly in the region of the trailing edge to reduce the centrifugal load. This result demonstrates that with



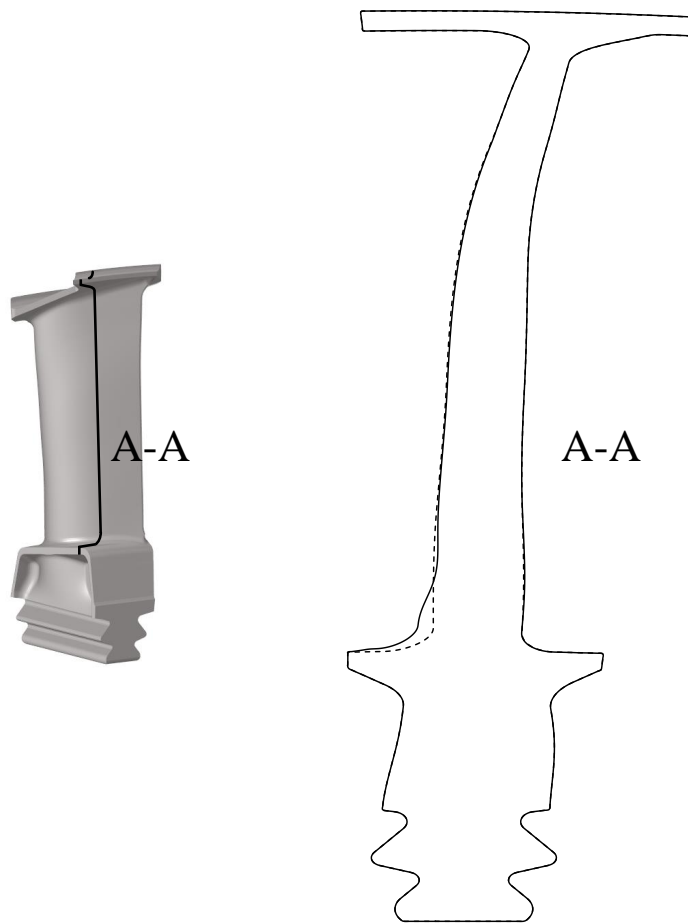
Vertex Morphing smooth design changes can take place while no wrinkles, kinks or any other mesh irregularities occur.



**Figure 4.19:** Surface sensitivity distribution of the stress objective function for the initial design.



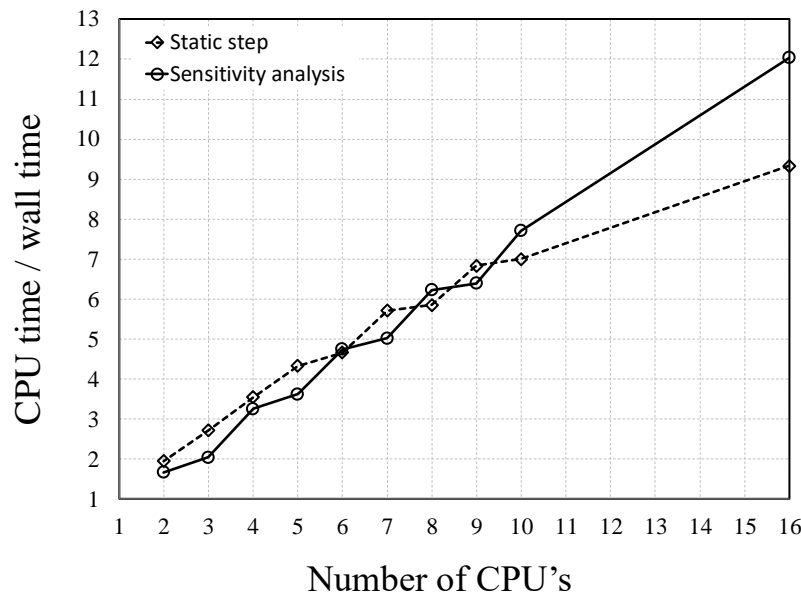
**Figure 4.20:** Cross section at the transition zone for the initial (dashed line) and the final design (continuous line).



**Figure 4.21:** Radial cross section of the initial (dashed line) and the final design (continuous line).

### 4.3.2 A time consumption study

Besides the development of meaningfully improved designs, the efficiency of the Vertex Morphing method with respect to the computational time is also of great interest for the application in an industrial environment. Therefore, the computational efficiency of the implemented sensitivity analysis is discussed and compared with the corresponding static analysis step. More precisely, the time dependency of the sensitivity analysis with respect to the number of CPU's, the number of design variables and the size objective set is investigated. The study is based on the low pressure turbine model shown in figure 4.14. The sensitivity analysis contains the computation of the gradients for the optimization problem defined in section 4.3.1, the filtering and the post-processing steps introduced in chapter 2.



**Figure 4.22:** Computational efficiency of the sensitivity analysis and the corresponding static step.

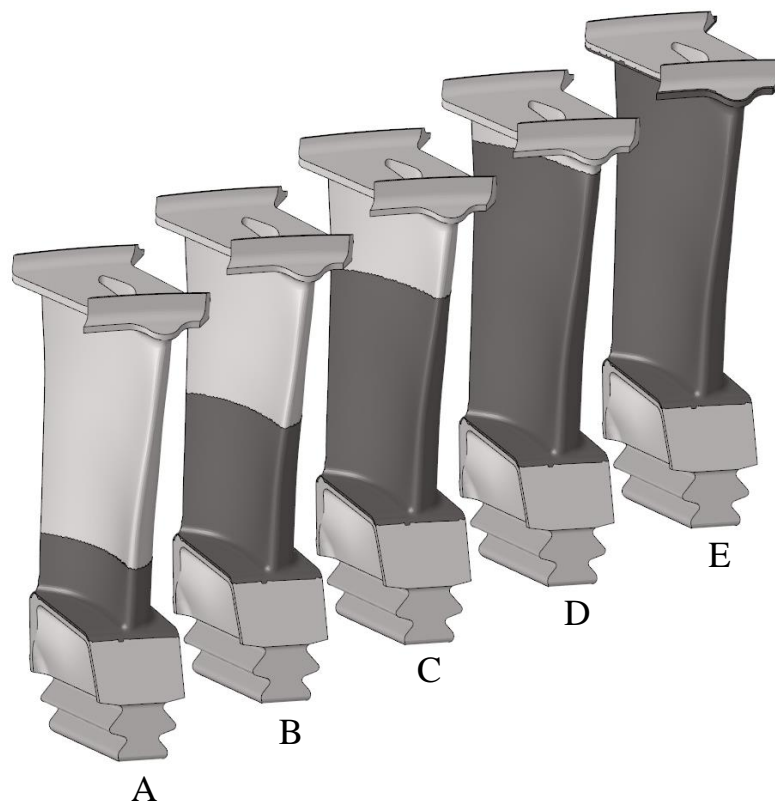
The most time consuming parts in the code like the partial derivatives in equation (2.16) or the filtering are parallelized. However, some parts still have to be executed sequentially. As a measure for the quality of the parallelization, the parallel efficiency is introduced which is defined as the ratio between the CPU time and the wall clock time. Ideally, the computational efficiency grows linearly with the number of CPU's with a slope of one meaning that the code is perfectly parallelized. Figure 4.22 compares the parallel efficiency of the static step and the sensitivity analysis implemented in CalculiX. For both, the behavior is approximately linear up to 16 CPU's. The slope of the static step is about 0.53 and the one of the sensitivity analysis step is approximately 0.75. From this results it can be concluded that the parallelization had been successfully implemented in CalculiX and the split of the problem to several threads clearly reduces the time consumption.

A second question that this study tries to answer is how influential is the size of the design space and the number of nodes evaluated in the objective function for the computational time of the sensitivity analysis. Therefore, several variants with different combinations of design spaces and sets for the objective function are defined in table 4.1 and visualized in figure 4.23. All other parameters which

influence the time consumption, e.g. the number of CPU's, are not varied in this study.

Variant	Number of design variables	Number of nodes in the objective function
A	26.939	26.939
B	52.913	52.913
C	73.487	73.487
D	93.330	93.330
E	113.702	113.702
F	52.913	26.939
G	73.487	26.939
H	93.330	26.939
I	113.702	26.939

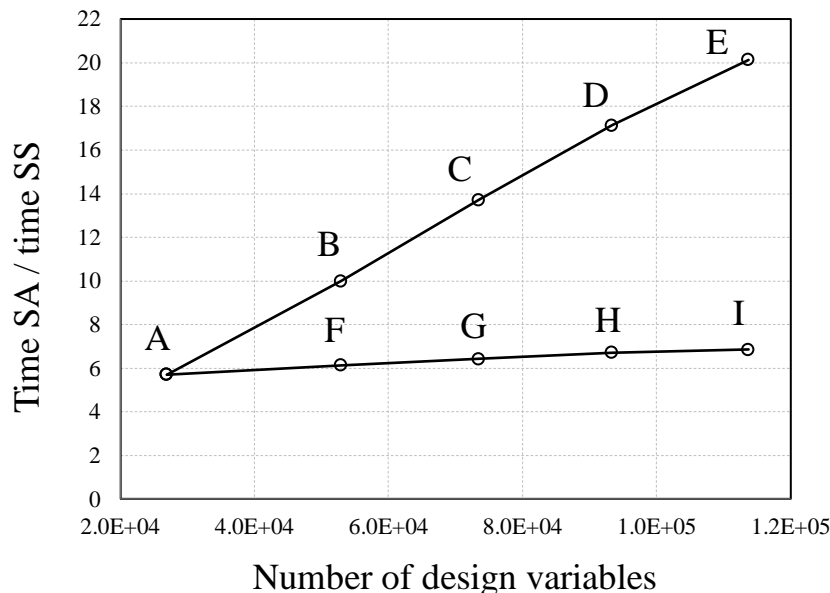
**Table 4.1:** Variants for parameter study with respect to the time consumption of the sensitivity analysis.



**Figure 4.23:** Different variants with respect to the design space.

Figure 4.24 evaluates the time consumption as the ratio between the time con-

sumption of the sensitivity analysis (SA) and the static step (SS). This value gives an indication how much more expensive a sensitivity step is compared to the solution of the underlying physical problem. For the variants *A* to *E*, the set where the objective function is evaluated is identical to the set of the design variables. In these cases the effort for the evaluation of the gradients compared to the effort for the solution of the static step increases linearly with an increasing number of design variables. In the variants *F* to *I*, the set for the objective function is kept constant (according to model *A* in figure 4.23) and only the size of the design space is changed. Similar to the variants *A* to *E*, the time consumption for the sensitivity analysis is also increasing for larger sets of design variables but the slope is less steep. The reduced effort is achieved by the screening the vectors  $\partial J/\partial \mathbf{s}$  and  $\partial J/\partial \mathbf{u}$  of equation (4.4) for zero entries as described in detail in section 3.5. This study verifies the idea how an examination of the mesh topology and especially the relationship between the design space and the set of the objective function can reduce the computational effort in many cases. It can also be concluded that a smart definition of the optimization improves the ratio between the time consumption of the sensitivity analysis and the static step.



**Figure 4.24:** Ratio of the computational time of the sensitivity analysis (SA) and the static step (SS) as a function of the number of design variables.

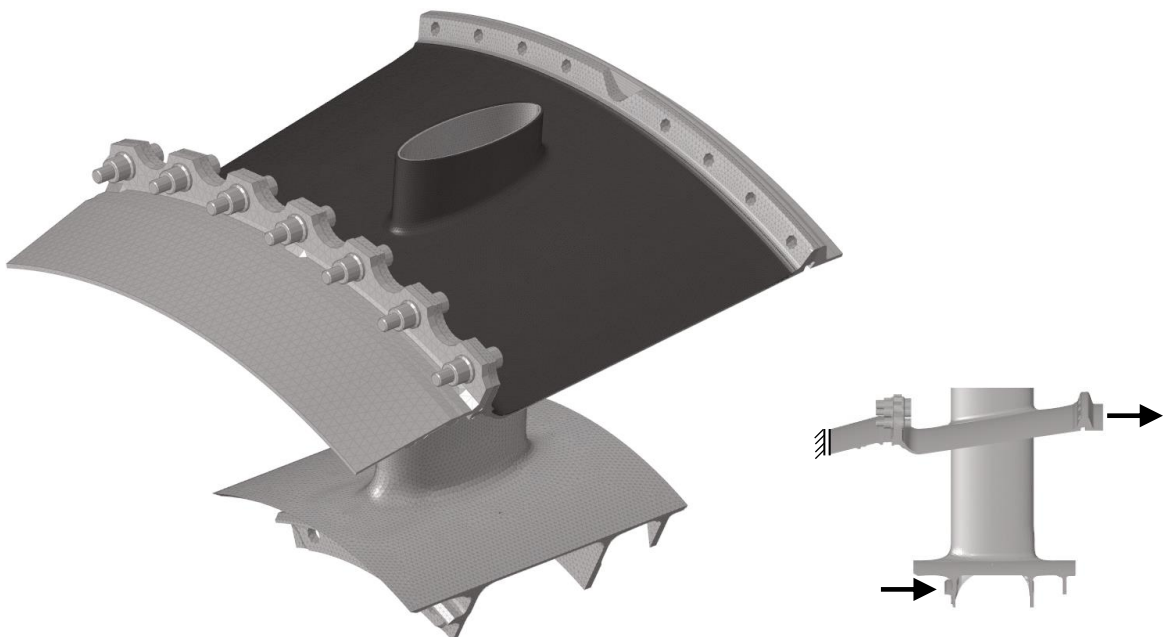
## 4.4 Case shell

In the current section the shape optimization of a case shell is examined discussing two important aspects in such industrial optimization problems. The first aspect is the influence of a geometrical wall thickness constraint on the optimization result. Generally, manufacturing of complex structures is many cases only possible by casting. For these kind of processes it is essential that the wall thickness remains within certain limits as too thick members have a negative effect on the cooling of the melt and too slender areas can not be filled with liquid. The result of the constrained optimization problem is presented and compared with a geometrically unconstrained optimization result. The second aspect discussed here is the influence of this geometrical constraint on the computational efficiency. As discussed in section 2.4 the geometrical constraint has to be defined for every design variable separately which potentially can lead to a huge number of active constraints in the computation of the feasible direction.

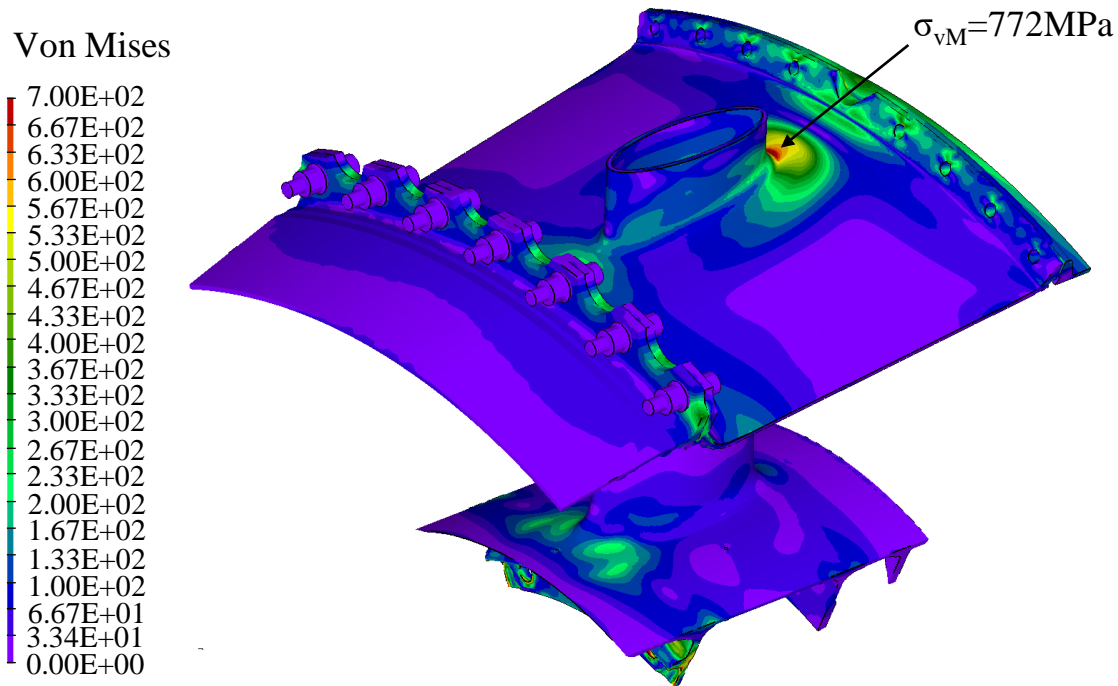


**Figure 4.25:** CAD model of engine component.

The outer case shell discussed here is part of a more complex structure which is illustrated in figure 4.25. The whole component is responsible to bypass the hot gas flow between the two cylindrical case shells in axial direction exposing it to high temperatures. In addition, the part has to transfer the loads from the bearing housing via the eight struts radially outwards to the engine mounts. This structure mechanical problem is solved in a linear elastic analysis on the basis of a  $45^\circ$  sector model with equivalent cyclic symmetry boundary conditions as shown in figure 4.26. According to the figure on the bottom-right the mechanical loads are applied on the inner as well as on the outer shell and the bearing is located on the left flange. The sector is discretized with 669.970 elements and 1.042.172 nodes. The design space which is marked in dark grey is highly resolved with second order tetrahedral elements and contains 136.800 design variables. The rest of the model consists of linear tetrahedral and quadrilateral elements. The combination of the mentioned mechanical and thermal loads leads to very high local stresses in the outer case shell. Figure 4.27 shows the corresponding von Mises stresses of the sector model highlighting the most critical position which occurs with 772 MPa in the transition of the case shell to the strut.



**Figure 4.26:** Discretized sector model of the engine component with the applied mechanical boundary conditions shown on the bottom-right.



**Figure 4.27:** Von Mises stresses of the initial design.

Goal of the shape optimization is the reduction of the von Mises stresses in the design space using the KS-function from equation (2.25) and repeated here

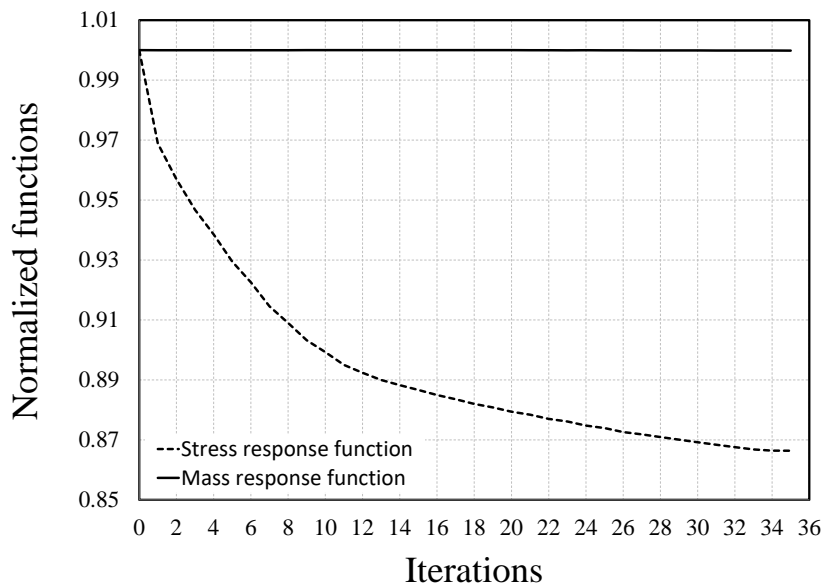
$$\sigma_{KS} = \frac{1}{\rho} \cdot \ln \left( \sum_{i=1}^{n_{nodes}} e^{\rho \cdot \frac{\sigma_i}{\bar{\sigma}}} \right). \quad (4.5)$$

The aggregation parameters  $\rho$  and  $\bar{\sigma}$  are set to 10 and 500 MPa, respectively. To avoid the trivial solution of reducing the stresses by just increasing the wall thickness, the side constraint that the mass has to remain below the initial value is added to the optimization problem.

In the Vertex Morphing optimization process, first the node-based sensitivity analysis for the stress objective function and the mass constraint is performed based on the results of the linear elastic analysis. Having the adjoint sensitivities available, the regularization of the gradient information is computed using the Gaussian kernel with a standard deviation which approximately spans nine elements in the filter radius. The Gradient Projection method uses the smoothed sensitivity fields to calculate a feasible descent direction by the projection the

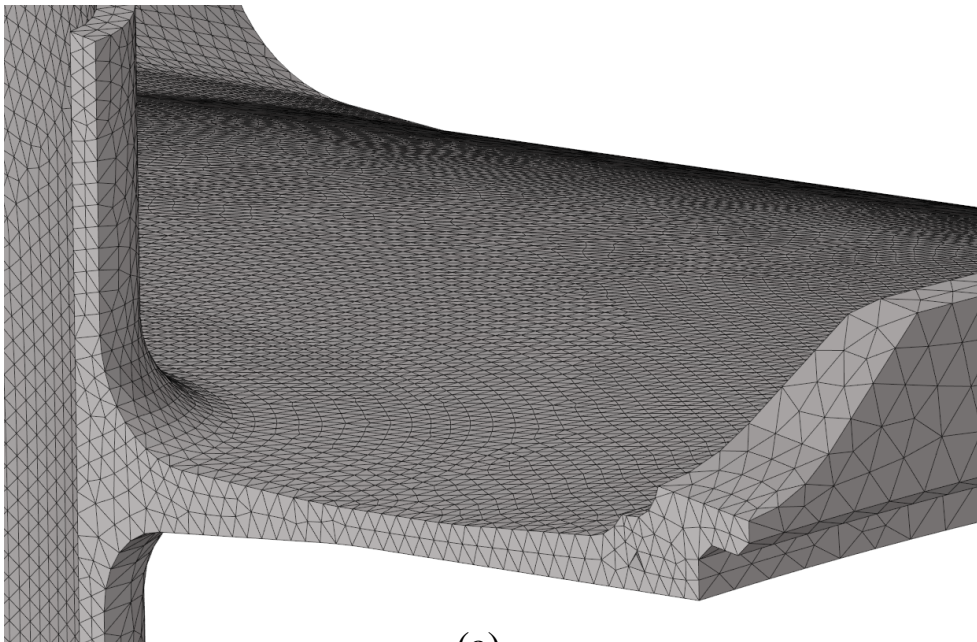


objective gradient on the subspace tangent of all active constraints. In the sequence, the mesh motion algorithm presented in section 2.6.5 updates the volume mesh on the basis of the shape update vector scaled with the predefined initial displacement of approximately 10 % of the average element length in the design space. For the preservation of the mesh quality of the elements next to the design variables, the number of element rows for the Young's modulus scaling is set to two. At the end of this iteration the structural assessment is repeated with the updated discretization. The loop is continued until the objective function has converged to a local minimum.

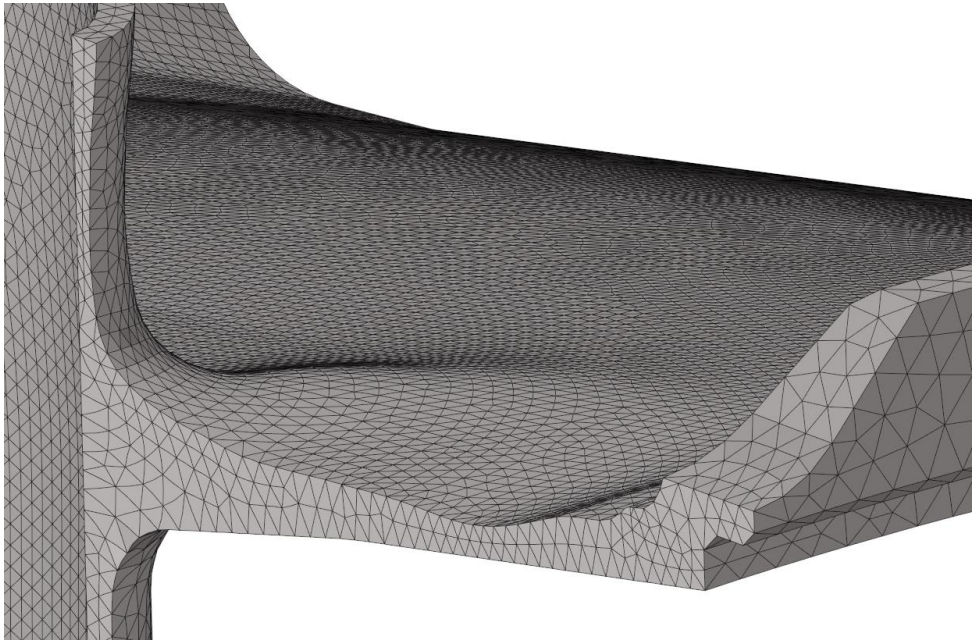


**Figure 4.28:** History for the normalized stress and mass response functions.

According to figure 4.28 the specific optimization problem converged after 35 iterations to a local optimum including a reduction of the stress objective function by 13 %. The resulting design which is shown in figure 4.29 reveals some interesting characteristics some of them being not evident at first sight. The radius in the transition zone where the highest stresses occur due to notch effects is clearly increased during the optimization process. In [Pet74] it is demonstrated on various examples that this is a common remedy for stress reduction in notched areas. The shell sections left and right from the strut are reinforced with smooth ribs to distribute the load more even in the case shell. In contrast to these features, the thickness of the shell section directly behind the strut had been



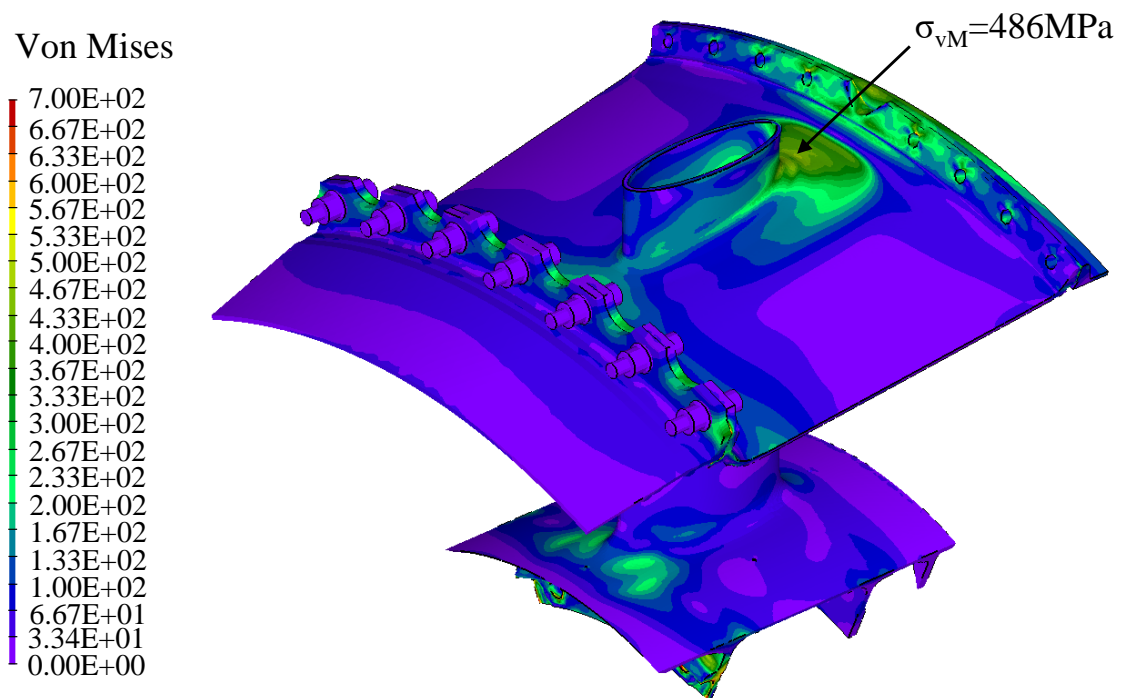
(a)



(b)

**Figure 4.29:** Shape optimization of case shell. Finite Element mesh of the initial design (a) and final design after convergence of the optimization algorithm (b).

reduced by the optimizer. The mechanical interpretation of all these changes can be that the stiffness in the transition zone is homogenized to distribute the load more even in the case shell. Figure 4.30 shows that the combination of all these single geometrical changes lead to a significant stress reduction of 37 % as in the final design a von Mises stress of only 486 MPa remains. Regarding the side constraint, figure 4.28 also shows that the new design is mass neutral compared to the initial one due to the fact that the constraint is kept at a value of approximately one by the Gradient Projection algorithm. In all the iterations the Lagrange multiplier is strictly positive indicating that the constraint is active during the whole optimization. The reduction of the shell thickness which compensates the additional mass of the ribs and the larger radius mainly takes place in the regions far away from the critical position. As the area which contributes to the mass compliance is very large, the absolute value of thickness reduction is comparably small.

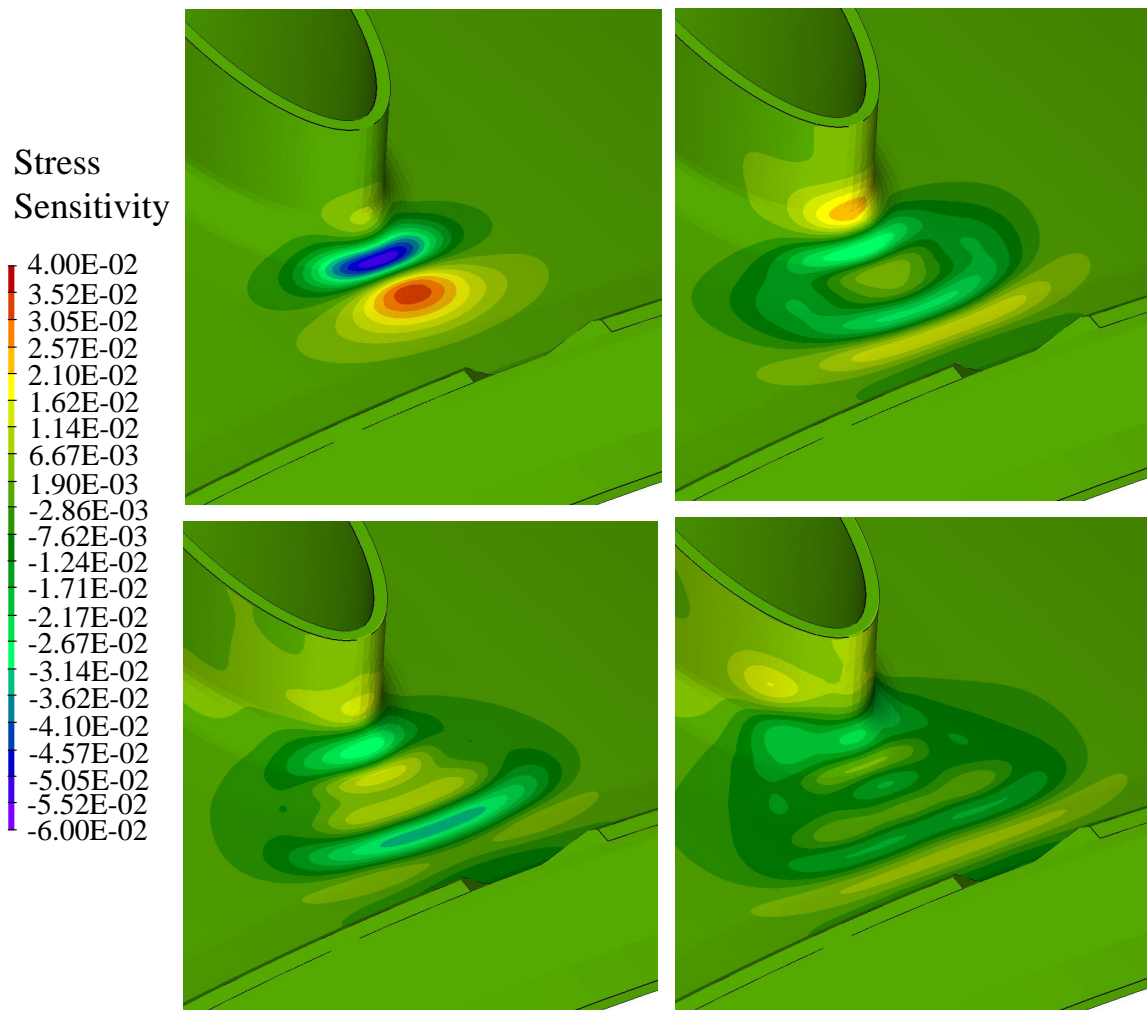


**Figure 4.30:** Von Mises stresses of the improved design.

The improvement of the design can also be seen through the sensitivity field of the objective function. Figure 4.31 displays the filtered surface sensitivity at different steps of the optimization. As shown in figure 4.31 at top-left, in the

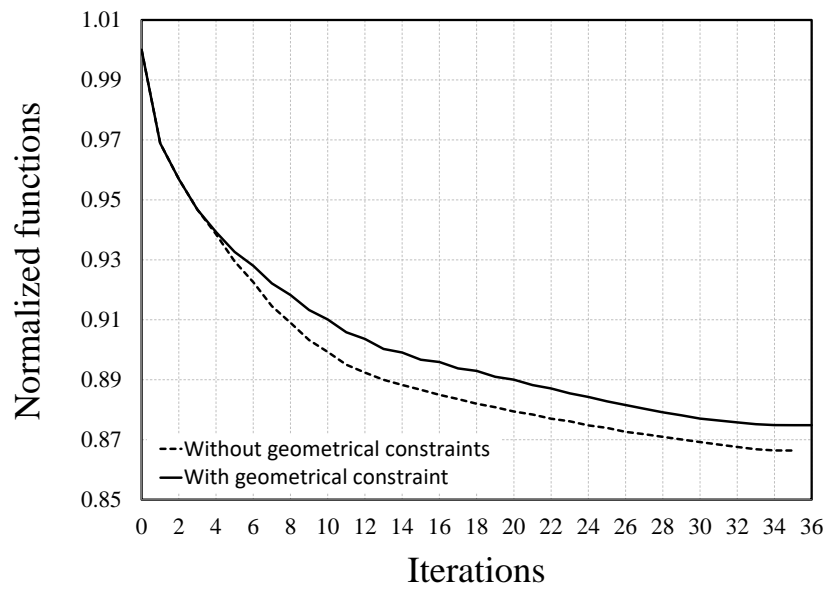
initial design the high sensitivities occur very local in the region of the highest stresses. In the course of the optimization when the peak stresses are eliminated and the stress field is more homogeneous the influence of other areas on the objective function increases leading to more evenly distributed sensitivities. As a general characteristic, the absolute values are decreasing towards the local optimum which also holds in this example. Theoretically, at a local optimum the gradients should even become zero. Here, in the final design which is shown in figure 4.31 at bottom-right the sensitivity values are considerably reduced but do not completely vanish. However, in constrained optimization problems the local optimum is often located in the infeasible region and can not be reached. In these cases the optimal design lies on the boundary of the active constraint called the constrained optimum. This case implies sensitivity values unequal to zero. In the example discussed here, the sensitivity information in the last iteration is unequal to zero but no design update could be computed which further improves the objective function. For any step length the Armijo condition can not be fulfilled thus the underlying constrained optimization problem is considered to be converged to the constrained optimum.

In order to show the necessity and the importance of geometric constraints in real size industrial applications the optimal design computed above is investigated in detail again with the focus on the resulting wall thickness. As aforementioned, to avoid problems in the manufacturing process the wall thickness often has to remain within certain limits. Comparing figure 4.29(a) and (b) it clearly can be seen that the vertical wall thickness of the strut has been reduced in the course of the optimization by 50 %. As the initial design is already borderline with respect to the minimum castable wall thickness the optimal design is not feasible taking this criterion into account. Therefore, the optimization is executed again with an additional geometrical constraint on the minimum wall thickness preventing a shrinkage below this limit. For the sake of confidentiality the dimensions and limits are not explicitly mentioned here but are not essential at all for the comprehension of the example. The geometrical constraint is defined according to section 2.4.2 requiring an additional control node set which is defined at the opposite sides of the shell regions. All the other boundary conditions remain

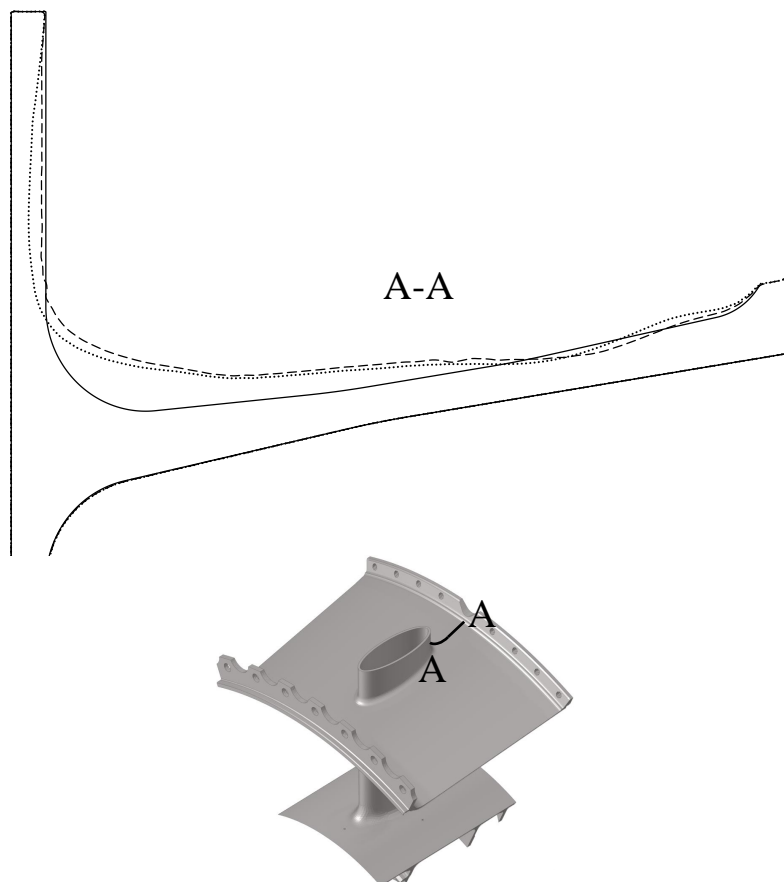


**Figure 4.31:** Sensitivity distribution of the stress objective function for the initial design at top-left, iteration 10 at top-right, iteration 20 at bottom-left and the optimal design at bottom-right.

unchanged. According to figure 4.32, after 36 iterations the problem converged to a local optimum. Compared to the optimization without any geometrical constraints the improvement of the objective function is approximately 1 % less. The corresponding von Mises stresses decreased to 521 MPa being 32.5 % of the initial design (compared to 37 % without any geometrical constraints). As expected, the additional constraints further restrict the feasible region and the optimal design is slightly worse than the one without any geometrical restrictions. Nevertheless, figure 4.33 showing the cross section of the region behind the strut demonstrates that the infeasible wall thickness shortfall can successfully be avoided.



**Figure 4.32:** Comparison of optimization history for the normalized stress response functions.



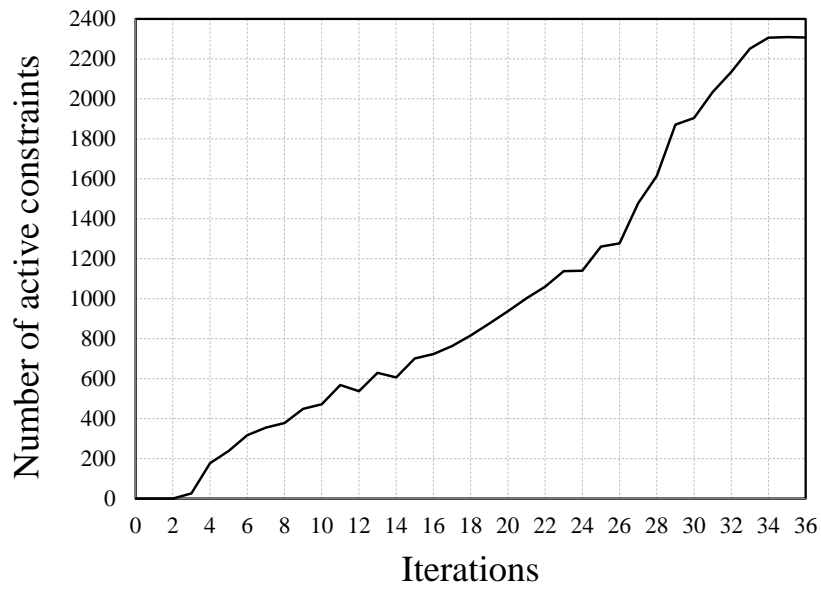
**Figure 4.33:** Comparison of cross sections of initial design (continuous line), geometrically unconstrained optimal design (dotted line) and optimal design with geometrical constraints (dashed line).

In section 2.4.2 it is stated that the geometrical constraint has to be defined separately for every design variable. Consequently, as many constraints as design variables could potentially enter the optimization problem which in case of real size industrial applications could lead to highly constrained optimization problems. For these kind of problems the preservation of the computational efficiency is a crucial aspect. In figure 4.34(a) the number of active geometrical constraints in every optimization iteration is shown for this specific example. Starting from an unconstrained optimization problem, the number of geometrical restrictions is steadily increasing up to 2307 active constraint in the final iteration. As stated in the equations (3.5), (3.6) and repeated here for clarity,

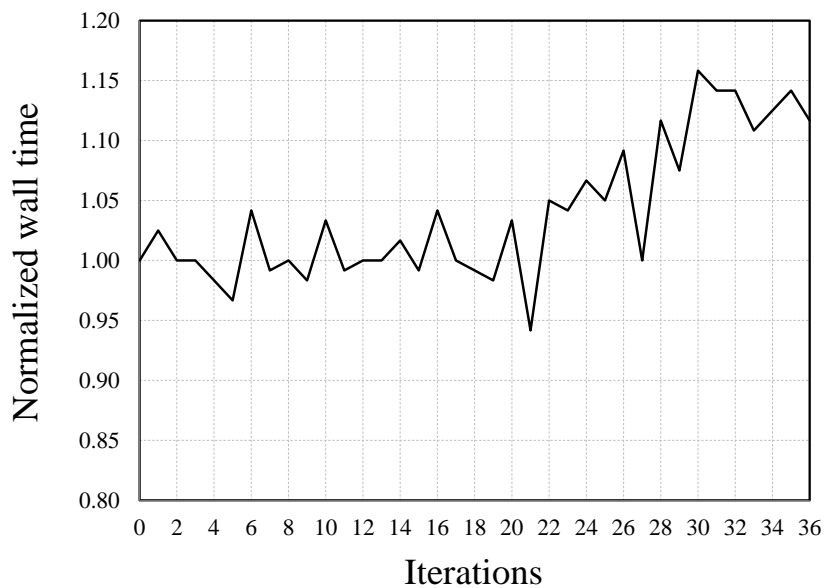
$$\mathbf{d}_n = -\mathbf{P}\nabla J, \quad (4.6)$$

$$\mathbf{P} = \mathbf{I} - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T, \quad (4.7)$$

in the Gradient Projection method the computation of the constrained descent direction  $\mathbf{d}_n$  is purely based on the  $\mathbf{N}$  matrix containing the gradient vectors of all active constraints. As  $\mathbf{N}$  has the size  $n \times n_{act}$ , for problems with many design variables which in addition are geometrically constrained too the matrix can quickly become very large. Especially the computation of the inverse of  $(\mathbf{N}^T\mathbf{N})^{-1}$  is very expensive. This demands for an efficient algorithm to compute the feasible descent direction  $(dJ/ds)_P$  derived in Appendix B. In figure 4.34(b) the time consumption for the computation of the feasible descent direction is evaluated. For the underlying problem, up to 1000 variables no remarkable increase in computational time can be detected. In the subsequent iterations the time consumption grows up to 115 % of the computation of the unconstrained descent direction. Considering the highly constrained optimization problem this additional effort seems to be reasonable. It also has to be mentioned that the algorithm is very robust in computing a feasible direction though a huge number of constraints is involved. Notice that the oscillations in the figure are not related to this investigation but account for the performance of the specific computer the iteration was executed on.



(a)



(b)

**Figure 4.34:** History of number of active constraints (a) and corresponding normalized wall clock time (normalization with respect to the time consumption for the computation of the unconstrained descent direction) (b).



# Conclusion and outlook

---

This contribution tried to take a step towards the integration of a numerical shape optimization tool in an industrial design process. The proposed approach is based on the *Vertex Morphing*, a node-based shape optimization method which efficiently transforms the result of the sensitivity analysis into meaningful designs by consistently incorporating the filter algorithm in the theory. The focus was on the following extensions of *Vertex Morphing*:

- Implementation of the sensitivity analysis for the relevant response functions in the open source Finite Element tool CalculiX.
- Regularization of the ill-posed shape optimization problem and the discretization.
- Consideration of constraints in the optimization algorithm reflecting given boundary conditions to attain feasible designs

In the course of this work the adjoint semi-analytic sensitivity analysis was successfully implemented in CalculiX. Herein, the adjoint manner accounts for the vast number of nodal design variables and the semi-analytic formulation avoids the cumbersome element specific derivation as the finite difference step is computed directly on element level. In structural optimization one is often

---

interested in mechanical stresses as they are directly linked to the life and load capability of a component. The gradient information for this important quantity is efficiently computed by the sophisticated Kreisselmeier-Steinhauser formulation which considers the influence of various nodal positions on the response function. Moreover, the influence of the discretization on the gradient information is described and the necessary remedies to achieve mesh independent results were presented and also integrated in the sensitivity analysis step.

The challenges of the node-based design variables were discussed in detail. At First, the ill-posed shape optimization problem needs to be regularized. In the *Vertex Morphing* method an additional design control field for the definition of the mathematical optimization problem is introduced. The connection between this control field and the nodal coordinates is established by a smoothing operator which transforms the noisy sensitivities into meaningful design updates. In contrast to other methods, this operation is consistently incorporated in the theory. The preservation of the quality of the three-dimensional mesh was the second challenge to be discussed. To get a global overview several mesh regularization methods were compared and the extended Traction method responsible for the mesh update in this work was presented. This approach was enriched by a distant dependent Young's modulus distribution and an inclusion of invariable areas in the design update further reducing the mesh deterioration and mesh dependencies.

In industrial applications shape optimal designs are typically constrained. Therefore, the optimization workflow was extended by Rosen's gradient projection method aggregating all active constraints into a feasible descent direction. The method projects the gradient of the objective function on the subspace tangent of all active constraints. Moreover, the decision which constraints have to be considered is controlled by an Active Set method.

Finally, the various methods were merged into a complete shape optimization workflow and its efficiency and robustness were demonstrated on several test cases. First, academic examples were examined to demonstrate accordance with

the analytically derived solution and the work of others. The second group of applications were from turbomachine industry and characterized by their geometrical complexity, large number of design variables and highly constrained design spaces.

To conclude, various publications already demonstrated the capabilities of *Vertex Morphing* as a powerful design tool offering a wide range of applications. In this work, the *Vertex Morphing* method was successfully extended for an application in a turbomachine design process. Therefore adequate response functions, geometrical constraints and the adjoint sensitivity analysis method were successfully implemented in CalculiX. The whole optimization workflow was completed by a constrained minimization algorithm and a robust mesh update step.

A worthwhile extension of the presented work would be the consideration of more than one load case in the optimization. Typically, aerospace components have to fulfill several load cases such as normal operating conditions, limit loads and ultimate loads. Often these requirements are contradicting and challenging to fulfill at the same time. The ability of the proposed workflow to consider additional load cases as side constraints preventing their violation would in fact be very beneficial. Moreover, to further increase the usability in industrial applications, one could also think of additional geometrical constraints such as rotational or cyclic symmetry.



---

# Transformation of partial derivative of the stress response function

---

Let us start from the partial derivative of the stress response function from equation (2.25) approximated by a finite difference step with respect to the design variables  $\mathbf{s}$

$$\begin{aligned}
 \frac{\partial \sigma_{KS}}{\partial s_i} &\approx \frac{\frac{1}{\rho} \cdot \ln(\mathcal{H} + \Delta \mathcal{H}) - \frac{1}{\rho} \cdot \ln(\mathcal{H})}{\Delta s_i} \\
 &\approx \frac{1}{\rho \cdot \Delta s_i} \cdot \ln(\mathcal{H} + \Delta \mathcal{H}) - \frac{1}{\rho \cdot \Delta s_i} \cdot \ln(\mathcal{H}) \\
 &\approx \frac{1}{\rho \cdot \Delta s_i} \cdot \ln \left[ \mathcal{H} \left( 1 + \frac{\Delta \mathcal{H}}{\mathcal{H}} \right) \right] - \frac{1}{\rho \cdot \Delta s_i} \cdot \ln(\mathcal{H}) \\
 &\approx \frac{1}{\rho \cdot \Delta s_i} \cdot \left[ \ln(\mathcal{H}) + \ln \left( 1 + \frac{\Delta \mathcal{H}}{\mathcal{H}} \right) - \ln(\mathcal{H}) \right] \\
 &\approx \frac{1}{\rho \cdot \Delta s_i} \cdot \ln \left( 1 + \frac{\Delta \mathcal{H}}{\mathcal{H}} \right)
 \end{aligned} \tag{A.1}$$

According to [BSMM05] the logarithm in equation (A.1) can be approximated by the Taylor series in the range from  $-1 < x < 1$

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots + (-1)^{n+1} \frac{x^n}{n} \pm \dots \tag{A.2}$$

The perturbation  $\Delta s_i$  is considered to be small compared to the dimensions of the model. Therefore, the higher-order terms of the Taylor series expansion are assumed to be of higher order small compared to the first-order term and thus neglected. With this assumption equation (A.1) reduces to

$$\frac{\partial \sigma_{KS}}{\partial s_i} \approx \frac{1}{\rho \cdot \Delta s_i} \cdot \frac{\Delta \mathcal{H}}{\mathcal{H}} \quad (\text{A.3})$$

After the simplification  $\Delta \mathcal{H}$  is substituted by the Kreisselmeier-Steinhauser summation

$$\begin{aligned} \frac{\partial \sigma_{KS}}{\partial s_i} &\approx \frac{1}{\rho \cdot \Delta s_i \cdot \mathcal{H}} \cdot \left( \sum_{i=1}^{n_{nodes}} e^{\rho \cdot \frac{\sigma_i + \Delta \sigma_i}{\bar{\sigma}}} - \sum_{i=1}^{n_{nodes}} e^{\rho \cdot \frac{\sigma_i}{\bar{\sigma}}} \right) \\ &\approx \frac{1}{\rho \cdot \Delta s_i \cdot \mathcal{H}} \cdot \left[ \sum_{i=1}^{n_{nodes}} e^{\rho \cdot \frac{\sigma_i}{\bar{\sigma}}} \left( e^{\rho \cdot \frac{\Delta \sigma_i}{\bar{\sigma}}} - 1 \right) \right] \end{aligned} \quad (\text{A.4})$$

Again, under the assumption of  $\Delta \sigma_i$  being small, equation (A.4) is further reduced with a Taylor series expansion of the exponential function according to [BSMM05] valid in the range from  $|x| < \infty$ .

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots \quad (\text{A.5})$$

Equation (A.5) is interrupted after the linear term and substituted in equation (A.4):

$$\frac{\partial \sigma_{KS}}{\partial s_i} \approx \frac{1}{\rho \cdot \Delta s_i \cdot \mathcal{H}} \cdot \left[ \sum_{i=1}^{n_{nodes}} e^{\rho \cdot \frac{\sigma_i}{\bar{\sigma}}} \left( 1 + \rho \cdot \frac{\Delta \sigma_i}{\bar{\sigma}} - 1 \right) \right]. \quad (\text{A.6})$$

The final expression implemented in the Finite Element code CalculiX reads as

$$\frac{\partial \sigma_{KS}}{\partial s_i} \approx \frac{1}{\bar{\sigma} \cdot \Delta s_i \cdot \mathcal{H}} \cdot \left( \sum_{i=1}^{n_{nodes}} e^{\rho \cdot \frac{\sigma_i}{\bar{\sigma}}} \cdot \Delta \sigma_i \right). \quad (\text{A.7})$$

---

# Efficient computation of projected gradient vector

---

The goal of the computation of the projected gradient vector  $\nabla_s J_{\mathbf{P}}$  is to take account of the constraints in the optimization problem. Therefore, the unconstrained gradient vector  $\nabla_s J$  of the objective function has to be projected onto the subspace tangent of all active constraints as follows:

$$\nabla_s J_{\mathbf{P}} = \mathbf{P}\nabla_s J = (\mathbf{I} - \mathbf{N}\mathbf{C}\mathbf{N}^T) \nabla_s J, \quad (\text{B.1})$$

with

$$\mathbf{C} = (\mathbf{N}^T \mathbf{N})^{-1}. \quad (\text{B.2})$$

In this equation  $\mathbf{P}$ ,  $\mathbf{N}$  and  $\mathbf{I}$  are the projection matrix, the matrix which contains the gradients of all active constraints and the identity matrix, respectively. The projected gradient can also be expressed in terms of the matrix coefficients,

$$\left( \frac{dJ}{ds_l} \right)_P = (\delta_{il} - n_{ij} c_{jk} (n^T)_{kl}) \frac{dJ}{ds_l}, \quad (\text{B.3})$$

and further transformed:

$$\begin{aligned}
\left(\frac{dJ}{ds_l}\right)_P &= (\delta_{il} - n_{ij}c_{jk}(n^T)_{kl}) \frac{dJ}{ds_l} &= \\
&= (\delta_{il} - n_{ij}c_{jk}n_{lk}) \frac{dJ}{ds_l} &= \\
&= \frac{dJ}{ds_i} - n_{ij}c_{jk}n_{lk} \frac{dJ}{ds_l} &= \\
&= \frac{dJ}{ds_i} - \sum_k \left[ n_{ij}c_{jk}n_{lk} \frac{dJ}{ds_l} \right] &= \\
&= \frac{dJ}{ds_i} - \sum_k \underbrace{\left[ \sum_j \sum_l \left( n_{ij}c_{jk}n_{lk} \frac{dJ}{ds_l} \right) \right]}_{w_{ik}} &=
\end{aligned} \tag{B.4}$$

In this equation the vector  $w_{ik}$  has to be computed  $k$  times with  $k = 1, \dots, n_{act}$  and is determined by the product of the vector  $v$  and the scalar value  $\beta$ :

$$w_{ik} = \underbrace{\begin{bmatrix} n_{11} & \cdots & n_{1n_{act}} \\ \vdots & \ddots & \vdots \\ n_{n_s,1} & \cdots & n_{n_s,n_{act}} \end{bmatrix}}_v \cdot \begin{bmatrix} c_{1k} \\ \vdots \\ c_{n_{act}k} \end{bmatrix} \cdot \underbrace{\begin{bmatrix} n_{1k} \\ \vdots \\ n_{n_s,k} \end{bmatrix}^T \begin{bmatrix} \frac{dJ}{ds_1} \\ \vdots \\ \frac{dJ}{ds_{n_s}} \end{bmatrix}}_{\beta} \tag{B.5}$$



---

# Bibliography

---

- [AFB10] ARNOUT, S. ; FURL, M. ; BLETZINGER, K.-U.: Parameter free optimization of shells considering stress response. In: *Proceeding of World Congress of Structural and Multidisciplinary Optimization* 9 (2010)
- [AGH01] AKGUN, M. ; GARCELON, J. ; HAFTKA, R.: Fast exact linear and non-linear structural reanalysis and the Sherman-Morrison-Woodbury formulas. In: *International Journal for Numerical Methods in Engineering* 50 (2001), 01, S. 1587 – 1606
- [AH73] AMSDEN, A. ; HIRT, C.: A Simple Scheme for Generating General Curvilinear Grids. In: *Journal of Computational Physics* 11 (1973), 03, S. 348–359
- [AKSK97] AZEGAMI, H. ; KAIZU, S. ; SHIMODA, M. ; KATAMINE, E.: Irregularity of shape optimization problems and an improvement technique. In: *International CONFERENCE on Computer Aided Optimum Design of Structures* (1997), 01
- [ALDDR10] ARNOUT, S. ; LOMBAERT, G. ; DEGRANDE, G. ; DE ROECK, G.: Optimization as a design tool for shell structures. In: *Proceeding of European Conference on Computational Mechanics* 4 (2010)
- [Ash06] ASHLOCK, D.: *Evolutionary computation for modeling and optimization*. Springer Verlag, 2006
- [AT06] AZEGAMI, H. ; TAKEUCHI, K.: A smoothing method for shape optimization: Traction method using the Robin condition. In:

- 
- International Journal of Computational Methods* 11 (2006), S. 21–33
- [AW96] AZEGAMI, H. ; WU, Z.: Domain Optimization Analysis in Linear Elastic Problems (Approach Using Traction Method). In: *JSME International Journal* 39 (1996), S. 272–278
- [AW05] ARORA, J. ; WANG, Q.: Review of formulations for structural and mechanical system optimization. In: *Structural and Multidisciplinary Optimization* 30 (2005), 01, S. 251–272
- [Ber99] BERTSEKAS, D.P.: *Nonlinear Programming*. Athena Scientific, 1999
- [BF84] BRAIBANT, V. ; FLEURY, C.: Shape optimal design using B-splines. In: *Computer Methods in Applied Mechanics and Engineering* 44 (1984), 08, S. 247–267
- [BFD08] BLETZINGER, K.-U. ; FIRL, M. ; DAOUD, F.: Approximation of derivatives in semi-analytical structural optimization. In: *Computers & Structures* 86 (2008), 07, S. 1404–1416
- [BFF10] BLETZINGER, K.-U. ; FIRL, M. ; FISCHER, M.: Parameter free shape design of thin shells: Efficient and effective, parallel solution techniques for very large design problems. In: *2<sup>nd</sup> International Conference on Engineering Optimization* (2010), 09
- [BH90] BARTHELEMY, B. ; HAFTKA, R.: Accuracy Analysis of the Semi-Analytical Method for Shape Sensitivity Calculation. In: *Mechanics of Structures and Machines* 18 (1990), 10
- [BK00] BOER, H. de ; KEULEN, F.: Refined semi-analytical design sensitivities. In: *International Journal of Solids and Structures* 37 (2000), 11, S. 6961–6980
- [Ble90] BLETZINGER, K.-U.: *Formoptimierung von Flächentragwerken*, University of Stuttgart, Diss., 1990

- [Ble98] BLETZINGER, K.-U.: *Form finding and optimization of membranes and minimal surfaces*. Lecture notes prepared for Ph.D.-course, 1998
- [Ble13] BLETZINGER, K.-U.: A consistent frame for sensitivity filtering and the vertex assigned morphing of optimal shape. In: *Structural Multidisciplinary Optimization* (2013)
- [BPBA99] BOUDAUD, A. ; PATRICIO, P. ; BEN AMAR, M.: The Helicoid versus the Catenoid: Geometrically Induced Bifurcations. In: *Physical Review Letters* 83 (1999), 11, S. 3836–3839
- [BR99] BLETZINGER, K.-U. ; RAMM, E.: A General Finite Element Approach to the Form Finding of Tensile Structures by the Updated Reference Strategy. In: *International Journal of Space Structures* 14 (1999), 06, S. 131–145
- [BS04] BENDSOE, M. P. ; SIGMUND, O.: *Topology Optimization: Theory, Methods and Applications*. Springer Verlag, 2004
- [BSB07] BOER, A. ; SCHOOT, M. ; BIJL, H.: Mesh deformation based on radial basis function interpolation. In: *Computers & Structures* 85 (2007), 06, S. 784–795
- [BSMM05] BRONSTEIN, I. N. ; SEMENDJAJEW, K. A. ; MUSIOL, G. ; MÜHLIG, H.: *Taschenbuch der Mathematik*. Bd. 6. Harri Deutsch, 2005
- [BW97] BONET, J. ; WOOD, R. D.: *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, 1997
- [BW01] BECKERT, A. ; WENDLAND, H.: Multivariate Interpolation for Fluid-Structure-Interaction Problems Using Radial Basis Functions. In: *Aerospace Science and Technology* 5 (2001), 03
- [BWDC05] BLETZINGER, K.-U. ; WÜCHNER, R. ; DAOUD, F. ; CAMPRUBI, N.: Computational methods for form finding and optimization

- of shells and membranes. In: *Computer Methods in Applied Mechanics and Engineering* 194 (2005), 08, S. 3438–3452
- [CK06] CHOI, K. K. ; KIM, N. H.: *Structural Sensitivity Analysis and Optimization 1: Linear Systems*. Springer Verlag, 2006
- [CO93] CHENG, G. ; OLHOFF, N.: New Method of Error Analysis and Detection in Semi-Analytical Sensitivity Analysis. 1 (1993), 01, S. 361–383
- [CSR87] CASTILLO, J. ; STEINBERG, S. ; ROACHE, P.: Mathematical aspects of variational grid generation II. In: *Journal of Computational and Applied Mathematics* 20 (1987), 11, S. 127–135
- [Dao05] DAOUD, F.: *Formoptimierung von Freiformschalen, Mathematische Algorithmen und Filtertechniken*, Technical University of Munich, Diss., 2005
- [DF02] DEGAND, C. ; FARHAT, C.: A three-dimensional torsional spring analogy method for unstructured dynamic meshes. In: *Computers & Structures* 80 (2002), 02, S. 305–316
- [Dho04] DHONDT, G.: *The Finite Element Method for Three-Dimensional Thermomechanical Applications*. John Wiley & Sons, 2004
- [Dho18] DHONDT, G.: *CalculiX CrunchiX USER'S MANUAL*. Bd. 2.15. 2018
- [DMSB99] DESBRUN, M. ; MEYER, M. ; SCHRÖDER, P. ; BARR, A.: Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow. In: *Computer Graphics SIGGRAPH* (1999), 09
- [EDB19] ERTL, F.-J. ; DHONDT, G. ; BLETZINGER, K.-U.: Vertex assigned morphing for parameter free shape optimization of 3-dimensional solid structures. In: *Computer Methods in Applied Mechanics and Engineering* 268 (2019), S. 494–513

- [EKO90] ESCHENAUER, H. ; KOSKI, J. ; OSYCZKA, A.: *Multicriteria Design Optimization - Procedures and Applications*. Springer Verlag, 1990
- [ELA11] ESPATH, L. ; LINN, R. ; AWRUCH, A.: Shape optimization of shell structures based on NURBS description using automatic differentiation. In: *International Journal for Numerical Methods in Engineering* 88 (2011), 11, S. 613 – 636
- [FB12] FIRL, M. ; BLETZINGER, K.-U.: Shape optimization of thin walled structures governed by geometrically nonlinear mechanics. In: *Computer Methods in Applied Mechanics and Engineering* 237–240 (2012), S. 107–117
- [FDKL98] FARHAT, C. ; DEGAND, C. ; KOOBUS, B. ; LESOINNE, M.: Torsional springs for two-dimensional dynamic unstructured fluid meshes. In: *Computer Methods in Applied Mechanics and Engineering* 163 (1998), 09, S. 231–245
- [Fir10] FIRL, M.: *Optimal Shape Design of Shell Structures*, Technical University of Munich, Diss., 2010
- [FWB13] FIRL, M. ; WÜCHNER, R. ; BLETZINGER, K.-U.: Regularization of shape optimization problems using FE-based parametrization. In: *Structural and Multidisciplinary Optimization* 47 (2013), 04
- [GH88] GOLDBERG, D. ; HOLLAND, J.: Genetic algorithms and machine learning. In: *Machine learning* 3 (1988), S. 95 – 99
- [GMW81] GILL, P. ; MURRAY, W. ; WRIGHT, M.: *Practical optimization*. Academic Press, 1981
- [HA89] HAFTKA, R. ; ADELMAN, H. M.: Recent developments in structural sensitivity analysis. In: *Structural Optimization* 1 (1989), 01, S. 137–151
- [Har08] HARZHEIM, L.: *Strukturoptimierung Grundlagen und Anwendungen*. Bd. 1. Harri Deutsch, 2008

- [HCT<sup>+</sup>99] HARDEE, E. ; CHANG, K.-H. ; TU, J. ; CHOI, K. ; GRINDEANU, I. ; YU, X.: A CAD-based design parameterization for shape optimization of elastic solids. In: *Advances in Engineering Software* 30 (1999), 03, S. 185–199
- [HDZ05] HANSEN, G. ; DOUGLASS, R. W. ; ZARDECKI, A.: *Mesh Enhancement: Selected Elliptic Methods, Foundations And Applications*. Imperial College Press, 2005
- [HG99] HAFTKA, R. ; GÜRDAL, Z.: *Elements of Structural Optimization. Third revised and expanded edition*. Kluwer Academic Publishers, 1999
- [HN88] HASLINGER, J. ; NEITTAANMÄKI, P.: *Finite element approximation for optimal shape design: theory and applications*. John Wiley & Sons, 1988
- [Hoj14] HOJJAT, M.: *Node-based parametrization for shape optimal design*, Technical University of Munich, Diss., 2014
- [HS52] HESTENES, M. R. ; STIEFEL, E.: Methods of Conjugate Gradients for Solving Linear Systems. In: *Journal of Research of the National Bureau of Standards* 49 (1952), 12
- [HSB14] HOJJAT, M. ; STAVROPOULOU, E. ; BLETZINGER, K.-U.: The Vertex morphing method for node-based shape optimization. In: *Computer Methods in Applied Mechanics and Engineering* 268 (2014), S. 494–513
- [Hug00] HUGHES, T. J. R.: *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Bd. 78. Dover Publications, 2000
- [HX96] HÄUSER, J. ; XIA, Y.: Modern Introduction to Grid Generation. In: *COSMASE Shortcourse Notes* (1996), 04

- [HZGB04] HANSEN, G. ; ZARDECKI, A. ; GREENING, D. ; BOS, R.: A finite element method for unstructured grid smoothing. In: *Journal of Computational Physics* 194 (2004), 03, S. 611–631
- [Ima82] IMAM, M.: Three dimensional shape optimization. In: *International Journal for Numerical Methods in Engineering* 18 (1982), 05, S. 661 – 673
- [INR07] I. N. ROZVANY, G.: Rozvany, G.I.N.: A Critical Review of Established Methods of Structural Topology Optimization. Structural and Multidisciplinary Optimization 37, 217-237. In: *Structural and Multidisciplinary Optimization* 37 (2007), 02, S. 217–237
- [Jia06] JIAO, Xiangmin: Face offsetting: A unified approach for explicit moving interfaces. In: *Journal of Computational Physics* 220 (2006), 08
- [JJ07] JAMESON, S. ; JAMESON, A.: Formulations for Topology, Shape and Discrete Optimization. In: *45th Aerospace Science Meeting and Exhibit* (2007)
- [JMP98] JAMESON, A. ; MARTINELLI, L. ; PIERCE, N.: Optimum aerodynamic design using the Navier-Stokes equations. In: *Theoretical and Computational Fluid Dynamics* 10 (1998), 01, S. 213–237
- [JT06] JASAK, H. ; TUKOVIC, Z.: Automatic mesh motion for the unstructured Finite Volume Method. In: *Transactions of FAMENA* 30 (2006), 11, S. 1–20
- [JV00] JAMESON, A. ; VASSBERG, J.: Studies of alternative numerical optimization methods applied to the Brachistochrone problem. In: *Computational Fluid Dynamics Journal* 9 (2000), S. 281–296
- [KHK05] KEULEN, F. ; HAFTKA, R. ; KIM, N.-H.: Review of options for structural design sensitivity analysis. Part 1: Linear systems. In: *Computer Methods in Applied Mechanics and Engineering* 194 (2005), 08, S. 3213–3243

- 
- [Kie11] KIENDL, J.: *Isogeometric Analysis and Shape Optimal Design of Shell Structures*, Technical University of Munich, Dissertation, 2011
- [Kir92] KIRSCH, U.: *Structural Optimization*. Springer Verlag, 1992
- [KL97] KIRSCH, U. ; LIU, S.: Structural Reanalysis for General Layout Modifications. In: *American Institute of Aeronautics and Astronautics 35* (1997), S. 382 – 388
- [Knu00] KNUPP, P.: Winslow Smoothing On Two-Dimensional Unstructured Meshes. (2000), 05
- [KS79] KREISSELMEIER, G. ; STEINHAUSER, R.: Systematic Control Design by Optimizing a Vector Performance Index. In: *IFAC Proceedings Volumes 12* (1979), S. 113–117
- [KS93] KNUPP, P. ; STEINBERG, S.: The Fundamentals of Grid Generation. (1993), 01
- [KSWB14] KIENDL, J. ; SCHMIDT, R. ; WÜCHNER, R. ; BLETZINGER, K.-U.: Isogeometric shape optimization of shells using semi-analytical sensitivity analysis and sensitivity weighting. In: *Computer Methods in Applied Mechanics and Engineering 274* (2014), S. 148–167
- [KT61] KUHN, H. ; TUCKER, A.: Nonlinear programming. In: *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability* (1961), S. 481 – 492
- [LB10] LINHARD, J. ; BLETZINGER, K.-U.: "Tracing" the equilibrium - Recent advances in numerical form finding. In: *International Journal of Space Structures 25* (2010), S. 107 – 116
- [LBT11] LE, C. ; BRUNS, T. ; TORTORELLI, D.: A gradient-based, parameter-free approach to shape optimization. In: *Computer Methods in Applied Mechanics and Engineering 200* (2011), S. 985–996



- [LCAT<sup>+</sup>17] LIAN, H. ; CHRISTIANSEN, N. A. ; A. TORTORELLI, D. ; SIGMUND, O. ; AAGE, N.: Combined shape and topology optimization for minimization of maximal von Mises stress. In: *Structural and Multidisciplinary Optimization* 5 (2017)
- [Lin94] LINDBERG, T.: *Scale-space theory in computer vision*. Kluwer Academic Publishers, 1994
- [Lin09] LINHARD, J.: *Numerisch-mechanische Betrachtung des Entwurfsprozesses von Membrantragwerken*, Technical University of Munich, Diss., 2009
- [LL86] LEE, D. ; LIN, A.: Generalized Delaunay triangulation for planar graphs. In: *Discrete & Computational Geometry* 1 (1986), 01, S. 201–217
- [LMGM05] LINGYUN, W. ; MEI, Z. ; GUANGMING, W. ; MENG, G.: Truss optimization on shape and sizing with frequency constraints based on genetic algorithm. In: *Computational Mechanics* 35 (2005), 04, S. 361–368
- [LO94] LUND, E. ; OLHOFF, N.: Shape design sensitivity analysis of eigenvalues using "exact" numerical differentiation of finite element matrices. In: *Structural Optimization* 8 (1994), 01, S. 52–59
- [LSS15] LIU, Y. ; SHIMODA, M. ; SHIBUTANI, Y.: Parameter-free method for the shape optimization of stiffeners on thin-walled structures to minimize stress concentration. In: *Journal of Mechanical Science and Technology* 29 (2015), 04, S. 1383–1390
- [Lue69] LUENBERGER, D.: *Optimization by vector space methods*. John Wiley & Sons, 1969
- [LY08] LUENBERGER, D. ; YE, Y.: *Linear and Nonlinear Programming*. Bd. 3. Springer Verlag, 2008

- [MAR04] MARTINS, J. ; ALONSO, J. ; REUTHER, J.: High-fidelity aerostructural design optimization of a supersonic business jet. In: *Journal of Aircraft* 41 (2004), S. 523–530
- [Mar09] MARÉCHAL, L.: Advances in Octree-Based All-Hexahedral Mesh Generation: Handling Sharp Features. In: *Proceedings of the 18th International Meshing Roundtable, IMR 2009* (2009), 01, S. 65–84
- [Moh97] MOHAMMADI, B.: A new optimal shape design procedure for inviscid and viscous turbulent flows. In: *International Journal for Numerical Methods in Fluids* 25 (1997), S. 183–203
- [MP01] MOHAMMADI, B. ; PIRONNEAU, O.: Applied shape optimization in fluids. In: *Applied Shape Optimization for Fluids* 10 (2001), 05, S. 213–237
- [MP09] MOHAMMADI, B. ; PIRONNEAU, O.: *Applied shape optimization for fluids*. Bd. 1. Numerical Mathematics and Scientific Computation. Oxford University Press, 2009
- [OBB01] OHTAKE, Y. ; BELYAEV, A. ; BOGAEVSKI, I.: Mesh regularization and adaptive smoothing. In: *Computer-Aided Design* 33 (2001), 09, S. 789–800
- [PCR89] PEDERSEN, P. ; CHENG, G. ; RASMUSSEN, J.: On Accuracy Problems for Semi-Analytical Sensitivity Analyses. In: *Mechanics of Structures and Machines* 17 (1989), S. 373–384
- [Pet74] PETERSON, R.: *Stress concentration factors*. John Wiley & Sons, 1974
- [Pir74] PIRONNEAU, O.: On optimum design in fluid mechanics. In: *Journal of Fluid Mechanics* 64 (1974), S. 97–110
- [PM07] POON, N.M.K. ; MARTINS, J.R.R.A.: An adaptive approach to constraint aggregation using adjoint sensitivity analysis. In: *Structural and Multidisciplinary Optimization* 34 (2007), S. 61–73

- [PTB<sup>+</sup>17] PICELLI, R. ; TOWNSEND, S. ; BRAMPTON, C. ; NORATO, J. ; KIM, H. A.: Stress-based shape and topology optimization with the level set method. In: *Computer Methods in Applied Mechanics and Engineering* 329 (2017), 10
- [QL10] QIU, Y. G. ; LI, X. S.: A note on the derivation of global stress constraints. In: *Structural and Multidisciplinary Optimization* 40 (2010), 01, S. 625–628
- [Rao09] RAO, S.: *Engineering Optimization: Theory and Practice*. Bd. 4. John Wiley & Sons, 2009
- [RFS<sup>+</sup>14] RIEHL, S. ; FRIEDERICH, J. ; SCHERER, M. ; MESKE, R. ; STEINMANN, P.: On the discrete variant of the traction method in parameter-free shape optimization. In: *Computer Methods in Applied Mechanics and Engineering* 278 (2014), S. 119–144
- [Ros60] ROSEN, J.B.: The Gradient Projection Method for Nonlinear Programming Part 1: Linear Constraints. In: *The Society for Industrial and Applied Mathematics*. 8 (1) (1960), S. 181–217
- [Ros61] ROSEN, J.B.: The Gradient Projection Method for Nonlinear Programming Part 2: Nonlinear Constraints. In: *The Society for Industrial and Applied Mathematics*. 9 (4) (1961), S. 514–532
- [Roz93] ROZVANY, G. I. N.: *Optimization of Large Structural Systems*. Bd. 1. Kluwer Academic Publishers, 1993
- [SCH00] SMITH, M. ; CESNIK, C. ; HODGES, D.: Evaluation of Some Data Transfer Algorithms for Noncontiguous Meshes. In: *Journal of Aerospace Engineering* 13 (2000), 04
- [Sch10] SCHMIDT, S.: *Efficient Large Scale Aerodynamic Design Based on Shape Calculus*, UNiversität Trier, Diss., 2010
- [Sch11] SCHERER, M.: *Regularizing Constraints for Mesh and Shape Optimization Problems*, University of Erlangen-Nürnberg, Diss., 06 2011

- [SHB14] STAVROPOULOU, E. ; HOJJAT, M. ; BLETZINGER, K.-U.: In-plane mesh regularization for node-based shape optimization problems. In: *Computer Methods in Applied Mechanics and Engineering* 275 (2014), 06, S. 39–54
- [She02] SHEWCHUK, J.: Delaunay Refinement Algorithms for Triangular Mesh Generation. In: *Computational Geometry* 22 (2002), 05, S. 21–74
- [SIGS08] SCHMIDT, S. ; ILIC, C. ; GAUGER, N. R. ; SCHULZ, V.: Shape gradients and their smoothness for practical aerodynamic design. In: *Optimization and Engineering, submitted (Preprint No. SPP1253-10-03)* (2008)
- [SL01] SOTO, O. ; LÖHNER, R.: CFD shape optimization using an incomplete-gradient adjoint formulation. In: *International Journal for Numerical Methods in Engineering* 51 (2001), 06, S. 735–753
- [SMA09] SHIMODA, M. ; MOTORA, S. ; AZEGAMI, H.: A practical solution to the shape optimization problem of solid structures. In: *Computer Aided Optimum Design in Engineering* 11 (2009), S. 33–44
- [SR11] STÜCK, A. ; RUNG, T.: Adjoint RANS with filtered shape derivatives for hydrodynamic optimisation. In: *Computers & Fluids* 47 (2011), S. 22–32
- [Sta15] STAVROPOULOU, E.: *Sensitivity analysis and regularization for shape optimization of coupled problems*, Technical University of Munich, Dissertation, 2015
- [Tau95] TAUBIN, G.: Curve and Surface Smoothing Without Shrinkage. In: *International Conference on Computer Vision* (1995), 07, S. 852–857
- [Tim51] TIMOSHENKO, S.: *The Theory of Elasticity*. Bd. 1. McGraw-Hill, 1951

- [TM94] TORTORELLI, D. ; MICHALERIS, P.: Design sensitivity analysis: Overview and review. In: *Inverse Problems in Engineering* 1 (1994), 10, S. 71–105
- [TSW99] THOMPSON, J. F. ; SONI, B. K. ; WEATHERILL, N. P.: *Handbook of grid generation*. CRC Press, 1999
- [Van17] VANDERPLAS, J.: *Python data science handbook*. O’Reilly, 2017
- [WB05] WÜCHNER, R. ; BLETZINGER, K.-U.: Stress-adapted numerical form finding of pre-stressed surfaces by the updated reference strategy. In: *International Journal for Numerical Methods in Engineering* 64 (2005), 09, S. 143 – 166
- [WCB15] WANG, W. ; CLAUSEN, P. ; BLETZINGER, K.-U.: Improved semi-analytical sensitivity analysis using a secant stiffness matrix for geometric nonlinear shape optimization. In: *Computers & Structures* 146 (2015), 01, S. 143–151
- [WCL18] WANG, G. ; CHEN, X. ; LIU, Z.: Mesh deformation on 3D complex configurations using multistep radial basis functions interpolation. In: *Chinese Journal of Aeronautics* 31 (2018), 02
- [WHR03] WALDMAN, W. ; HELLER, M. ; ROSE, L.: Shape optimisation of two closely-spaced holes for fatigue life extension. In: *DSTO Platforms Sciences Laboratory* (2003)
- [Wri01] WRIGGERS, P.: *Nichtlineare Finite-Element-Methoden*. Springer Verlag, 2001
- [Yos95] YOSIDA, K.: *Functional Analysis*. Springer Verlag, 1995
- [Zou60] ZOUTENDIJK, G.: Methods of feasible directions. In: *Elsevier* (1960)



---

# Bisherige Titel der Schriftenreihe

---

## Band Titel

- 1 Frank Koschnick, *Geometrische Lockingeffekte bei Finiten Elementen und ein allgemeines Konzept zu ihrer Vermeidung*, 2004.
- 2 Natalia Camprubi, *Design and Analysis in Shape Optimization of Shells*, 2004.
- 3 Bernhard Thomée, *Physikalisch nichtlineare Berechnung von Stahlfaserbetonkonstruktionen*, 2005.
- 4 Fernaß Daoud, *Formoptimierung von Freiformschalen - Mathematische Algorithmen und Filtertechniken*, 2005.
- 5 Manfred Bischoff, *Models and Finite Elements for Thin-walled Structures*, 2005.
- 6 Alexander Hörmann, *Ermittlung optimierter Stabwerkmodelle auf Basis des Kraftflusses als Anwendung plattformunabhängiger Prozesskopplung*, 2006.
- 7 Roland Wüchner, *Mechanik und Numerik der Formfindung und Fluid-Struktur-Interaktion von Membrantragwerken*, 2006.
- 8 Florian Jurecka, *Robust Design Optimization Based on Metamodeling Techniques*, 2007.
- 9 Johannes Linhard, *Numerisch-mechanische Betrachtung des Entwurfsprozesses von Membrantragwerken*, 2009.
- 10 Alexander Kupzok, *Modeling the Interaction of Wind and Membrane Structures by Numerical Simulation*, 2009.
- 11 Bin Yang, *Modified Particle Swarm Optimizers and their Application to Robust Design and Structural Optimization*, 2009.

**Band Titel**

- 12 Michael Fleischer, *Absicherung der virtuellen Prozesskette für Folgeoperationen in der Umformtechnik*, 2009.
- 13 Amphon Jrusjrungkiat, *Nonlinear Analysis of Pneumatic Membranes - From Subgrid to Interface*, 2009.
- 14 Alexander Michalski, *Simulation leichter Flächentragwerke in einer numerisch generierten atmosphärischen Grenzschicht*, 2010.
- 15 Matthias Firl, *Optimal Shape Design of Shell Structures*, 2010.
- 16 Thomas Gallinger, *Effiziente Algorithmen zur partitionierten Lösung stark gekoppelter Probleme der Fluid-Struktur-Wechselwirkung*, 2011.
- 17 Josef Kiendl, *Isogeometric Analysis and Shape Optimal Design of Shell Structures*, 2011.
- 18 Joseph Jordan, *Effiziente Simulation großer Mauerwerksstrukturen mit diskreten Rissmodellen*, 2011.
- 19 Albrecht von Boetticher, *Flexible Hangmurenbarrieren: Eine numerische Modellierung des Tragwerks, der Hangmure und der Fluid-Struktur-Interaktion*, 2012.
- 20 Robert Schmidt, *Trimming, Mapping, and Optimization in Isogeometric Analysis of Shell Structures*, 2013.
- 21 Michael Fischer, *Finite Element Based Simulation, Design and Control of Piezoelectric and Lightweight Smart Structures*, 2013.
- 22 Falko Hartmut Dieringer, *Numerical Methods for the Design and Analysis for Tensile Structures*, 2014.
- 23 Rupert Fisch, *Code Verification of Partitioned FSI Environments for Lightweight Structures*, 2014.
- 24 Stefan Sicklinger, *Stabilized Co-Simulation of Coupled Problems Including Fields and Signals*, 2014.
- 25 Majid Hojjat, *Node-based parametrization for shape optimal design*, 2015.
- 26 Ute Israel, *Optimierung in der Fluid-Struktur-Interaktion - Sensitivitätsanalyse für die Formoptimierung auf Grundlage des partitionierten Verfahrens*, 2015.



**Band Titel**

- 27 Electra Stavropoulou, *Sensitivity analysis and regularization for shape optimization of coupled problems*, 2015.
- 28 Daniel Markus, *Numerical and Experimental Modeling for Shape Optimization of Offshore Structures*, 2015.
- 29 Pablo Suárez, *Design Process for the Shape Optimization of Pressurized Bulkheads as Components of Aircraft Structures*, 2015.
- 30 Armin Widhammer, *Variation of Reference Strategy - Generation of Optimized Cutting Patterns for Textile Fabrics*, 2015.
- 31 Helmut Masching, *Parameter Free Optimization of Shape Adaptive Shell Structures*, 2016.
- 32 Hao Zhang, *A General Approach for Solving Inverse Problems in Geophysical Systems by Applying Finite Element Method and Metamodel Techniques*, 2016.
- 33 Tianyang Wang, *Development of Co-Simulation Environment and Mapping Algorithms*, 2016.
- 34 Michael Breitenberger, *CAD-integrated Design and Analysis of Shell Structures*, 2016.
- 35 Önay Can, *Functional Adaptation with Hyperkinematics using Natural Element Method: Application for Articular Cartilage*, 2016.
- 36 Benedikt Philipp, *Methodological Treatment of Non-linear Structural Behavior in the Design, Analysis and Verification of Lightweight Structures*, 2017.
- 37 Michael Andre, *Aeroelastic Modeling and Simulation for the Assessment of Wind Effects on a Parabolic Trough Solar Collector*, 2018.
- 38 Andreas Apostolatos, *Isogeometric Analysis of Thin-Walled Structures on Multipatch Surfaces in Fluid-Structure Interaction*, 2018.
- 39 Altuğ Emiroğlu, *Multiphysics Simulation and CAD-Integrated Shape Optimization in Fluid-Structure Interaction*, 2019.
- 40 Mehran Saeedi, *Multi-Fidelity Aeroelastic Analysis of Flexible Membrane Wind Turbine Blades*, 2017.

**Band Titel**

- 41 Reza Najian Asl, *Shape Optimization and Sensitivity Analysis of Fluids, Structures, and their Interaction Using Vertex Morphing Parametrization*, 2019.
- 42 Ahmed Abodonya, *Verification Methodology for Computational Wind Engineering Prediction of Wind Loads on Structures*, 2020.
- 43 Anna Maria Bauer, *CAD-Integrated Isogeometric Analysis and Design of Lightweight Structures*, 2020.
- 44 Andreas Winterstein, *Modeling and Simulation of Wind-Structure Interaction of Slender Civil Engineering Structures Including Vibration Mitigation Systems*, 2020.