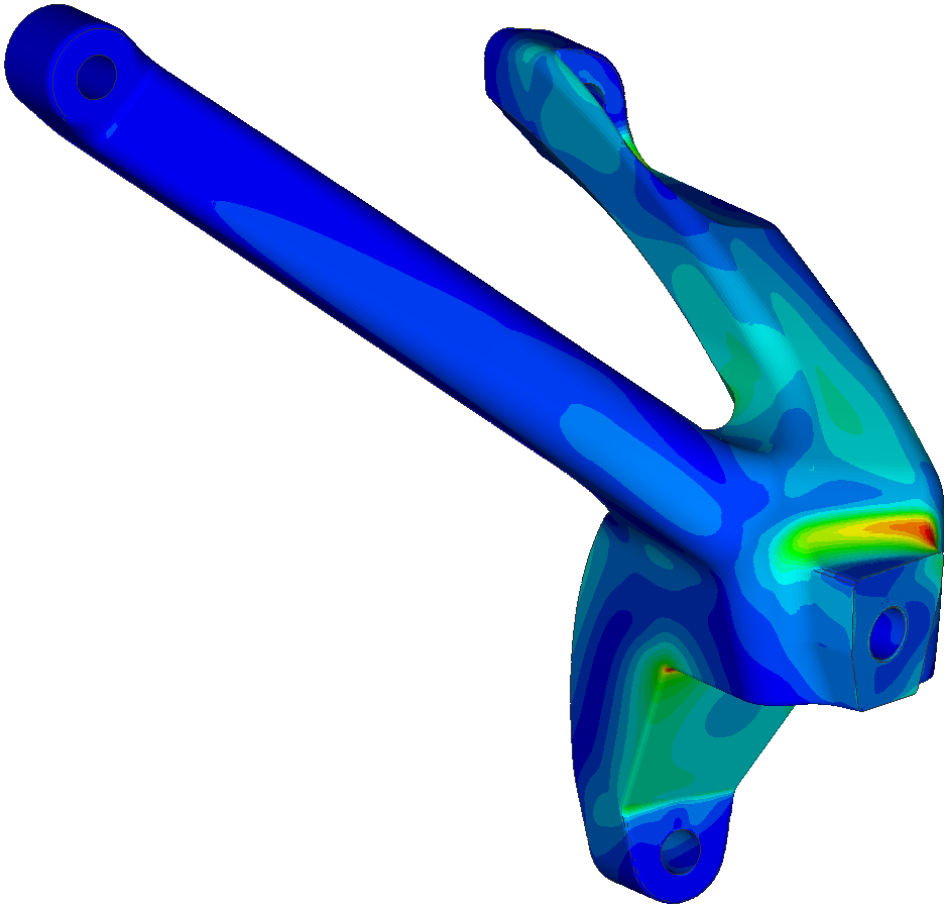
CHALMERS



Structural Optimization of Components for High-Performance Engines

Master's thesis in Applied Mechanics

MICHAEL SCHRECK

Department of Applied Mechanics
Division of Material and Computational Mechanics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2014
Master's thesis 2014:56

MASTER'S THESIS IN APPLIED MECHANICS

Structural Optimization of Components for High-Performance Engines

MICHAEL SCHRECK

Department of Applied Mechanics
Division of Material and Computational Mechanics
CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2014

Structural Optimization of Components for High-Performance Engines
MICHAEL SCHRECK

Master's thesis 2014:56
ISSN 1652-8557
Department of Applied Mechanics
Division of Material and Computational Mechanics
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-772 1000

Cover:
Stress analysis of a weight-optimized engine mount bracket

Chalmers Reproservice
Göteborg, Sweden 2014

Structural Optimization of Components for High-Performance Engines
Master's thesis in Applied Mechanics
MICHAEL SCHRECK
Department of Applied Mechanics
Division of Material and Computational Mechanics
Chalmers University of Technology

Abstract

In the course of modern vehicle development, the demands for shorter development periods and reduced costs are constantly increasing. Therefore, virtual development using numerical computer tools plays a significant role to achieve these goals. In particular, numerical structural optimization allows a goal-oriented design of engine components from the very beginning of product development. Given a design space, suitable boundary conditions and load cases, a rough first design draft of a component can be derived by means of a topology optimization. A subsequent shape optimization then finalizes the design by eliminating structural weak points that could cause failure.

In this paper, a weight-optimized design of an engine mount bracket for a V8 engine is proposed and compared to the current design. Both topology and shape optimizations are performed using the commercial finite element software PERMAS and OptiStruct, respectively. The design goal of topology optimization is to produce an equally stiff component at reduced structural weight compared to the existing design. The objective of a subsequent shape optimization addresses a stress reduction such that operational strength under multi-axial and cyclic operating loading is proven. For this purpose, the fatigue life prediction software FEMFAT is used. Concluding modal and buckling analyses ensure uncritical vibrational and buckling behaviour of the new engine mount bracket.

The design derived from topology optimization reaches stiffness comparable to the existing component while the total weight could be reduced by 23 %. The most critical stress concentration compromising operational strength could be removed during shape optimization. Moreover, the first eigenfrequency of the engine was slightly reduced by the new design and there is no danger of buckling for compressive loads.

Throughout this paper, essential knowledge about optimization methods and how to use them in day-to-day tasks is acquired. Recommendations for further use in the calculation/simulation department of Dr. Ing. h.c. F. Porsche AG are made, along with a preview of new optimization possibilities in upcoming software releases.

Keywords: Structural optimization, Topology optimization, Shape optimization, Multi-axial fatigue analysis, Engine mount bracket, PERMAS, OptiStruct, FEMFAT

Zusammenfassung

Im Zuge moderner Fahrzeugentwicklung steigen die Ansprüche an verkürzte Entwicklungszeiten und reduzierte Kosten stetig an. Aus diesem Grund spielt die virtuelle Entwicklung, die sich numerischer Berechnungstools bedient, eine immer wichtigere Rolle zur Erreichung dieser Ziele. Insbesondere die numerische Strukturoptimierung erlaubt eine zielorientierte Auslegung von Motorkomponenten von Beginn der Produktentwicklung an. Unter Angabe eines Designraumes, passender Randbedingungen und Lasten kann mittels einer Topologieoptimierung ein schnelles, erstes Designkonzept abgeleitet werden. Eine darauffolgende Formoptimierung, deren Ziel die Beseitigung von versagensrelevanten Schwachstellen ist, schließt den Designprozess ab.
In dieser Arbeit wird ein gewichtsoptimierter Entwurf für einen Motortragarm eines V8-Motors erstellt und mit dem derzeit existierenden Bauteil verglichen. Sowohl Topologie- als auch Formoptimierungen werden hierbei mit der kommerziellen Finite-Elemente-Software PERMAS und OptiStruct durchgeführt. Das Ziel der Topologieoptimierung ist ein Bauteil zu erstellen, das im Vergleich zum Bestehenden die gleiche Steifigkeit bei reduziertem Bauteilgewicht besitzt. Die darauffolgende Formoptimierung zielt auf eine Spannungsreduzierung ab, so dass Betriebsfestigkeit für die multi-axialen und zyklischen Betriebslasten gewährleistet ist. Diese Bewertung erfolgt mittels der Software FEMFAT. Abschließende Modal- und Knick-Analysen sichern das unkritische Schwing- und Knickverhalten des neuen Motortragarms ab.
Das aus der Topologieoptimierung abgeleitete Bauteildesign erreicht eine mit dem bestehenden Bauteil vergleichbare Steifigkeit wobei das Gesamtgewicht um 23 % reduziert werden konnte. Die kritischste Spannungsspitze, die die Betriebsfestigkeit gefährdet, konnte während der Formoptimierung entfernt werden. Die Eigenfrequenz des Halbmotormodells verringerte sich geringfügig, weiterhin besteht keine Knickgefahr bei auftretenden Drucklasten.
Im Zuge dieser Arbeit wurde grundlegendes Wissen über Optimierungsmethoden und deren Anwendung in alltäglichen Aufgabenstellungen gesammelt. Empfehlungen hinsichtlich des weiteren Gebrauchs in der Berechnungs- & Simulationsabteilung der Dr. Ing h.c. F. Porsche AG wurden erarbeitet sowie ein Ausblick auf Entwicklungen in kommenden Software-Releases erstellt.

Schlüsselwörter: Strukturoptimierung, Topologieoptimierung, Formoptimierung, Multi-axiale Ermüdungsanalyse, Motortragarm, PERMAS, OptiStruct, FEMFAT

Acknowledgements

The work presented in this thesis report was conducted at the Porsche Weissach Development Centre in Germany. I owe my deepest gratitude to those who made the outcome of this work possible.

First and foremost, I would like to thank my supervisor Natalia Herth (Dr. Ing. h.c. F. Porsche AG) for her continuous support and helpfulness throughout this thesis. Her qualified inputs contributed to the quality of this work to a large extent and helped me solve the issues that arose during these six months.

I would like to express my very great appreciation to my examiner Prof. Lennart Josefson (Chalmers University of Technology) for his valuable and constructive critiques of this research work. His willingness to enable a thesis project abroad and to give his time so generously has been very much appreciated.

I would also like thank a number of individuals at Dr. Ing. h.c. F. Porsche AG and INTES GmbH who helped and supported me to fulfil my goals:

- Thomas Conze, Senior engineer in the power train calculations department, for his patient support in all matters of finite element calculations and his helpful advices.
- Dr. Stefan Schwarz, Senior engineer in the body calculations department, for providing profound knowledge on structural optimization and the use of Altair OptiStruct.
- Jens Müller and all other colleagues from INTES GmbH for their continuous support for the finite element software PERMAS.
- Senior engineers Thomas Riebl and Joachim Henn and intern Patrick Leidig from the design department, for providing essential information on the engine mount bracket and creating CAD models.
- Gary Avery, Senior engineer in the power train department, for his generous support to improve the quality of this thesis.

Finally, I wish to thank all my other colleagues in the power train department at Porsche and the department of Applied Mechanics at Chalmers University of Technology for treating me as one of you.

Michael Schreck

Göteborg, Sweden 2014

List of Abbreviations and Acronyms

AUTO	**AUTO**matic
BFGS	**B**royden-**F**letcher-**G**oldfarb-**S**hanno
CAD	**C**omputer-**A**ided **D**esign
CAE	**C**omputer-**A**ided **E**ngineering
CAO	**C**omputer-**A**ided **O**ptimization
CPU	**C**entral **P**rocessing **U**nit
DOF	**D**egree **O**f **F**reedom
DV	**D**esign **V**ariable
ESO	**E**volutionary **S**tructural **O**ptimization
CONLIN	**CON**vex **LIN**earization
FE	**F**inite **El**ement
FEM	**F**inite **El**ement **M**ethod
GB	**G**iga**B**yte
GCA	**G**eneralized **C**onvex **A**pproximation
GPU	**G**raphics **P**rocessing **U**nit
KKT	**K**arush-**K**uhn-**T**ucker
LDR	**L**ocally improved **D**i**R**ect scheme
MMA	**M**ethod of **M**oving **A**symptotes
MPC	**M**ulti **P**oint **C**onstraint
MS	**M**ulti**S**tart
NVH	**N**oise **V**ibration **H**arshness
RBE	**R**igid **B**ody **E**lement
SCP	**S**equential **C**onvex **P**rogramming
SCPS	**S**equential **C**onvex **P**rogramming with **S**parse constraints
SIMP	**S**olid **I**sotropic **M**aterial with **P**enalization
SLP	**S**equential **L**inear **P**rogramming
SLPS	**S**equential **L**inear **P**rogramming with **S**parse constraints
SPC	**S**ingle **P**oint **C**onstraint
SQP	**S**equential **Q**uadratic **P**rogramming
SQPS	**S**equential **Q**uadratic **P**rogramming with **S**parse constraints
SRSM	**S**uccessive **R**esponse **S**urface **M**ethod
STL	**ST**ereo**L**ithography
WLIN	**W**edge-constrained **LIN**ear programming

Nomenclature

Scalar quantities

b	$[-]$	Fatigue strength exponent
	$[-]$	Slope of S-N curve
c	$[-]$	Fatigue ductility exponent
C	$[\mathrm{N} \cdot \mathrm{m}]$	Global elastic strain energy (frequently denoted "compliance")
E	$[\mathrm{N} \cdot \mathrm{m}^{-2}]$	Elastic Young's modulus
ER	$[-]$	Evolutionary rate
$f(\mathbf{x})$	$[-]$	Continuous objective function
F	$[\mathrm{N}]$	Force
$g_j(\mathbf{x})$	$[-]$	Continuous inequality constraint functions
$h_k(\mathbf{x})$	$[-]$	Continuous equality constraint functions
i	$[-]$	Imaginary unit: $i^2 = -1$
k_i	$[-]$	Weight factor
K'	$[-]$	Cyclic hardening coefficient
m	$[\mathrm{kg}]$	Mass
m_0	$[\mathrm{kg}]$	Reference mass
n'	$[-]$	Cyclic hardening exponent
n_{dv}	$[-]$	Number of design variables
N_f	$[-]$	Cyclic limit of endurance
p	$[-]$	Penalty factor
R	$[-]$	Stress ratio
R_i	$[\mathrm{N}]$	Reaction forces
RR	$[-]$	Rejection ratio
S	$[\mathrm{N} \cdot \mathrm{m}^{-2}]$	Nominal stress
S_f	$[\mathrm{N} \cdot \mathrm{m}^{-2}]$	Stress limit of endurance
SF	$[-]$	Safety factor
SF_A	$[-]$	Endurance safety factor
t	$[\mathrm{s}]$	Time
V_j	$[\mathrm{m}^3]$	Element volume
x_i	$[-]$	Element filling ratio
${\varepsilon'}_f$	$[-]$	Fatigue ductility coefficient
γ	$[-]$	Lagrange multiplier assigned to equality constraint function
$\mathcal{L}$	$[-]$	Lagrangian
λ	$[-]$	Lagrange multiplier assigned to inequality constraint function

ν	$[-]$	Elastic Poisson's ratio
ρ	$[\mathrm{kg} \cdot \mathrm{m}^{-3}]$	Mass density
ω_0	$[\mathrm{s}^{-1}]$	Eigenfrequency
σ_a	$[\mathrm{N} \cdot \mathrm{m}^{-2}]$	Amplitude stress
σ'_f	$[\mathrm{N} \cdot \mathrm{m}^{-2}]$	Fatigue strength coefficient
σ_m	$[\mathrm{N} \cdot \mathrm{m}^{-2}]$	Mean stress
σ_u	$[\mathrm{N} \cdot \mathrm{m}^{-2}]$	Ultimate strength
σ_{vM}	$[\mathrm{N} \cdot \mathrm{m}^{-2}]$	Von Mises equivalent stress
σ_y	$[\mathrm{N} \cdot \mathrm{m}^{-2}]$	Yield strength

Vectors, first order tensors

$\mathbf{f}$	$[\mathrm{N}]$	Force vector
$\mathbf{u}$	$[\mathrm{m}]$	Displacement vector
$\dot{\mathbf{u}}$	$[\mathrm{m} \cdot \mathrm{s}^{-1}]$	Velocity vector
$\ddot{\mathbf{u}}$	$[\mathrm{m} \cdot \mathrm{s}^{-2}]$	Acceleration vector
$\mathbf{x}$	$[-]$	Real vector of design variables
$\mathbf{x}^*$	$[-]$	Global solution to the optimization problem

Matrices, second and higher order tensors

$\boldsymbol{C}$	$[\mathrm{N} \cdot \mathrm{m}^{-2}]$	Elasticity tensor
$\mathbf{E}$	$[\mathrm{N} \cdot \mathrm{m}^{-2}]$	Element stiffness matrix
$\mathbf{H}(f)$	$[-]$	Hessian matrix of a function f
$\mathbf{K}$	$[\mathrm{N} \cdot \mathrm{m}^{-1}]$	Global elastic stiffness matrix
$\mathbf{K}_0$	$[\mathrm{N} \cdot \mathrm{m}^{-1}]$	Base state stiffness matrix
$\mathbf{K}_\Delta$	$[\mathrm{N} \cdot \mathrm{m}^{-1}]$	Load stiffness matrix
$\mathbf{M}$	$[\mathrm{kg}]$	Global mass matrix
$\boldsymbol{\varepsilon}$	$[-]$	Cauchy strain tensor
$\boldsymbol{\rho}_i$	$[\mathrm{kg} \cdot \mathrm{m}^{-3}]$	Element mass density matrix
$\boldsymbol{\sigma}$	$[\mathrm{N} \cdot \mathrm{m}^{-2}]$	Cauchy stress tensor

Contents

1 Introduction

"The art of structure is where to put the holes."

Robert Le Ricolais (1894-1977)

In an age when development time and costs are more important than ever, CAE (computer-aided engineering) plays an important role alongside traditional testing. Long before the first prototype is built, virtual simulations can reveal meaningful insights into the functioning and possible malfunctioning of a product [1].

One particular area that can be applied throughout the whole development process, is structural optimization. This mathematical method aims on an optimized design of a desired structure while satisfying reasonable constraints such as maximum weight or stresses. The term "optimized" refers to a defined goal function that could be structural stiffness or any other mechanical quantity of interest. Structural optimization is divided into three classes: topology, shape and sizing optimization [2].

Topology optimization, as the most general method, aims to find the basic distribution of material in the so-called design space - the available space that can be occupied by the component. In automotive engineering, this design space is usually restricted by neighbouring components or other design constraints in a vehicle. This method attempts to predict a design that can withstand the defined loads on the component in the best possible way. The most common design goal is structural stiffness for obvious reasons [3]. Clearly, topology optimization is a useful tool if there is little experience about the mechanical behaviour of a component but it also allows to compare existing designs to a topology optimization and possibly identify potential for improvement.

Shape optimization, the second and less general method, is restricted to modifying only the outer surface (boundary) of a component but is not capable of creating new topologies such as voids. It can be applied as subsequent step of a topology optimization or it can be used to simply optimize an already existing component that shows critical system responses, e.g. elevated stress or unacceptable deformations. Thus, shape optimization is only applied locally where a structure has found to be critical with the risk of undesired effects.

The third and least general method, sizing optimization, modifies (scalar) design parameters such as thickness, cross-section or material parameters to improve a component's ability to sustain certain loads. Usually, this method is applied for analyses of one- or two-dimensional models with rod, bar or plate elements which is not covered here.

Over the recent years, numerical structural optimization has become increasingly popular as it has been implemented in popular commercial finite element packages such as Abaqus, ANSYS, Nastran or OptiStruct. This trend has also been supported by the rapid development of computing power which enables the efficient and economical use of costly numerical optimization.

However, many new challenges and questions have to be addressed when using this method. Therefore, a profound understanding of the basics of numerical structural optimization as well as its application in combination with finite element analysis is essential.

1.1 Purpose

The purpose of the present work is to provide the company with deeper knowledge about the use of structural optimization methods, both in theory and practice using commercial software. This is achieved by a trial study on idealized topology and shape optimization problems. The matters of interest concern the modelling using different finite elements with different refinements, the influence of boundary conditions and load cases, the solution methods and solver settings in general.
On this basis, a complete structural optimization of an engine mount bracket is performed and evaluated on the basis of an existing design concept. Finite element analyses are complemented by a multi-axial fatigue analysis for the calculation of operational strength.
During the course of this project, a guideline for later optimization projects at the company is to be formulated.

1.2 Objective

In accordance with the above, the objectives are formulated as follows:

- Literature research (theoretical assessment and evaluation of different optimization methods)
- Analysis of available optimization tools in the finite element software PERMAS[1] by means of an academic example. Analysis of influence factors: finite element modelling, boundary conditions and loads, optimization goal and constraint functions and more
- Application of topology and shape optimization on a highly stressed engine component using PERMAS and subsequent transfer to CAD
- Consideration of (PERMAS-external) fatigue analysis results for optimization within PERMAS
- Formulation of guidelines for the integration of numerical structural optimization in the process chain of engine development

1.3 Scope

The present work focuses on structural optimization using commercial code that is used in the calculation & simulation department of Porsche. Due to time constraints, no experimental testing on obtained design drafts can be evaluated and compared to numerical results.
As regards content, gathered knowledge about governing parameters and dependencies are presented and interpreted. Apart from introductory trial cases, a real engine mount is optimized from scratch to final.

1.4 Outline of the thesis

This report is divided into six main chapters. After this introduction, Chapter 2 gives an overview of the theoretical background of structural optimization and how to use commercial software for numerical optimization. Chapter 3 then describes the methodological part of this work, addressing all necessary steps when performing structural optimization with the finite element software PERMAS and OptiStruct, respectively. All obtained results and their discussion are covered in Chapter 4.
Finally, this report is concluded with a summary in Chapter 5 and recommendations for further investigations in Chapter 6.

[1] PERMAS is a finite element software that has been developed by the company INTES GmbH, headquartered in Stuttgart, Germany, since 1984.

2 Theoretical Framework

In this chapter the theoretical background of this work is presented. First, the fundamentals of structural optimization are outlined and solution methods as well as challenges are addressed. Further on, solving optimization problems with commercial finite element software is treated, based on day-to-day tasks as performed in industry.

2.1 Fundamentals of structural optimization

The subject of structural optimization is to find a mechanical structure of assembled material that sustains defined loads in the best possible way. The term "best" refers to a mechanical criterion that can be evaluated numerically. As an example, a typical goal is to make a structure as light as possible while satisfying stress limits or to design it as stiff as possible for a given weight limit. In structural optimization, these goals and constraints are formulated in mathematical expressions that have to be solved for the unknown parameters. The theoretical background according to standard references [1, 2, 4, 5, 6, 7] is treated briefly here.

2.1.1 Mathematical formulation

Mathematically, an optimization problem addresses the determination of a solution set $\mathbf{x} \in \mathbb{R}^n$ that minimizes (or maximizes[1]) a (scalar) function $f : \mathbb{R}^n \to \mathbb{R}$. In addition, constraints functions $g_j(\mathbf{x})$ and $h_k(\mathbf{x})$ on its variables $\mathbf{x}$ can be stated, as outlined in Eq. 2.1. The optimization problem herein can thus be written in the following general form:

$$\begin{array}{lllr} \text{minimize} & f(\mathbf{x}) & , & \mathbf{x} \in \mathbb{R}^n \\ \text{subject to} & g_j(\mathbf{x}) \leq 0 & , & j = 1, \ldots, m \\ & h_k(\mathbf{x}) = 0 & , & k = 1, \ldots, p \end{array} \tag{2.1}$$

In Eq. 2.1, the following notation is used:

- $\mathbf{x} \in \mathbb{R}^n$: a real vector of n design variables (also unknowns, parameters),
- $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$: a continuous goal function (also objective or cost function),
- $g_j(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^m$: continuous inequality constraint functions,
- $h_k(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^p$: continuous equality constraint functions.

The vector $\mathbf{x}^* \in \mathbb{R}^n$ that minimizes the goal function $f(\mathbf{x})$ inside the entire domain while satisfying the constraint conditions in $g_j(\mathbf{x})$ and $h_k(\mathbf{x})$ is called the *global solution* or *global minimizer*. Thus,

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad , \quad \forall\, \mathbf{x} \in \mathbb{R}^n \tag{2.2}$$

By contrast, a vector $\mathbf{x} \in \mathbb{S}^n$ that minimizes $f(\mathbf{x})$ only inside an arbitrarily small neighbourhood $\mathbb{S}^n$ of $\mathbf{x}$ but not for the entire domain $\mathbb{R}^n$ is called a *local solution* or *local minimizer*. For a local minimum, it can be stated

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad , \quad \forall\, \mathbf{x} \in \mathbb{S}^n \subset \mathbb{R}^n \tag{2.3}$$

Only for a convex optimization problem with $f(\mathbf{x}), g(\mathbf{x})$ and $h(\mathbf{x})$ all being convex functions, the local solution is identical to the global solution [8, p. 152]. Otherwise, this is not necessarily true. The problem of local minima for non-convex problems will be addressed later in Chapter 2.1.8.1.

[1] The maximization of a function f is equivalent to the minimization of the function $-f$. However, only the case of minimization will be considered in the sequel for convenience.

2.1.2 Lagrange duality

The mathematical analysis of the optimization problem with separate constraint equations (Eq. 2.1) can be quite cumbersome. Usually, it is easier to study an optimization problem with joint objective and constraint functions. Such a formulation can be achieved by using the *Lagrange function* [9, p. 385]. The main idea is to incooperate a weighted sum of the constraint equations. The weight factors are referred to as *Lagrange multipliers* λ_j and ν_k. The so-called *Lagrangian* $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ is then defined as:

$$\begin{aligned} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\gamma}) &= f(\mathbf{x}) + \sum_{j=1}^{m} \lambda_j g_j(\mathbf{x}) + \sum_{k=1}^{p} \gamma_k h_k(\mathbf{x}) \\ \text{with} \quad & \mathbf{x} \in \mathbb{R}^n, \quad \boldsymbol{\lambda} \in \mathbb{R}^m, \quad \boldsymbol{\gamma} \in \mathbb{R}^p, \quad \lambda_j, \gamma_k \geq 0. \end{aligned} \tag{2.4}$$

The initial optimization problem from Eq. 2.1 is then stated with the *Lagrange dual function* $g : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ which is the minimum of the Lagrangian:

$$g(\lambda, \gamma) = \inf_{\mathbf{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\gamma}) = \inf_{\mathbf{x}} \left(f(\mathbf{x}) + \sum_{j=1}^{m} \lambda_j g_j(\mathbf{x}) + \sum_{k=1}^{p} \gamma_k h_k(\mathbf{x}) \right) \tag{2.5}$$

2.1.3 Optimality conditions

Optimality conditions are conditions for a solution $\mathbf{x}^*$ to be optimal. The necessary optimality KKT condition and a sufficient optimality condition will be outlined briefly in the sequel.

2.1.3.1 Necessary optimality conditions (Karush-Kuhn-Tucker conditions)

The so-called *Karush-Kuhn-Tucker* (KKT) conditions [8, p. 243] are first order necessary conditions that must be satisfied for any convex or non-convex optimization in order for $\mathbf{x}^*$ to be the optimal solution:
Since the primal solution $\mathbf{x}^*$ minimizes the Lagrangian $\mathcal{L}$, its gradient $\nabla\mathcal{L}$ must be zero, i.e.

$$\nabla\mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\gamma}^*) = \nabla f(\mathbf{x}^*) + \sum_{j=1}^{m} \lambda_j^* \nabla g_j(\mathbf{x}^*) + \sum_{k=1}^{p} \gamma_k^* \nabla h_k(\mathbf{x}^*) = 0 \tag{2.6}$$

Thus, the following conditions for $\lambda_j, \gamma_k \geq 0$ must be satisfied

$$\begin{aligned} g_j(\mathbf{x}^*) &\leq 0 \quad , \quad j = 1, \ldots, \mathrm{m} \\ h_k(\mathbf{x}^*) &= 0 \quad , \quad k = 1, \ldots, \mathrm{p} \\ \lambda_j^* \nabla g_j(\mathbf{x}^*) &= 0 \quad , \quad j = 1, \ldots, \mathrm{m} \\ \nabla\mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\gamma}^*) &= 0 \quad , \quad j = 1, \ldots, \mathrm{m} \ , \ k = 1, \ldots, \mathrm{p} \end{aligned} \tag{2.7}$$

The first two conditions in Eq. 2.7 are the inequality and equality constraints itself that must be fulfilled. The third term accounts for the fact that the weighted sum of all g_j in the Lagrangian must vanish for the optimal point. The last condition is the requirement that the gradient of the Lagrangian must vanish. The primal optimal point $\mathbf{x}^*$ and dual optimal point $(\boldsymbol{\lambda}^*, \boldsymbol{\gamma}^*)$ are also called the *KKT point*.

2.1.3.2 Sufficient optimality condition

If an optimization problem is convex, then a KKT point (see Ch. 2.1.3.1) is automatically the global optimum. In the special case of unconstrained optimization, a single sufficient condition for a global optimum is that the gradient $\boldsymbol{\nabla} f(\mathbf{x}^*) = \mathbf{0}$ vanishes [9, p. 385]. It may be noted that the solution of a constrained problem might also lie on the boundary of the feasible set, i.e. the subset $\mathbf{x} \subset \mathbb{R}^n$ where all constraint equations are fulfilled.

2.1.4 Sensitivity analysis

The sensitivity analysis covers the calculation of derivatives with respect to $\mathbf{x}$ which are needed to obtain the solution $\mathbf{x}^*$, i.e.

$$\begin{aligned} &\boldsymbol{\nabla} f(\mathbf{x}) \\ &\boldsymbol{\nabla} g_j(\mathbf{x}) \quad , \quad j = 1, \ldots, m \\ &\boldsymbol{\nabla} h_k(\mathbf{x}) \quad , \quad k = 1, \ldots, p \end{aligned} \tag{2.8}$$

The determination of these gradients is of great importance since it can take up most of the time during on optimization. The implementation of the following methods in the finite element software PERMAS is described in Ch. 2.2.2.3 and Ch 2.2.3.3 for topology and shape optimization, respectively. Generally, there a two major approaches to calculate the terms in Eq. 2.8: finite differences or (semi-)analytical methods which use either the direct method or the adjoint method.

2.1.4.1 Finite differences

In the simplest case, the gradients are approximated by calculating finite differences for n design variables whereby forward differences[2] are employed here:

$$\frac{\partial f(\mathbf{x})}{\partial x_i} \approx \frac{\Delta f(\mathbf{x})}{\Delta x_i} = \frac{f(x_1, x_2, \ldots, x_{i-1}, x_i + \Delta x_i, x_{i+1}, \ldots, x_n) + f(x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)}{\Delta x_i} \tag{2.9}$$

However, this method requires that $n + 1$ finite differences have to be calculated in total which is very costly for a big number of design variables.
Regarding accuracy, there are two sources of error when using finite differences. The *truncation error* arises from neglecting higher-order terms of the Taylor series expansion in Eq. 2.9. This error can be reduced by small perturbations Δx_i. A small perturbation, however, can cause a high *contribution error* due to computational round-off errors. This goal conflict is referred to as the *step-size dilemma* [4, p. 15].

2.1.4.2 Semi-analytical and analytical methods

The motivation for using analytical or semi-analytical methods instead of finite differences is to provide a more efficient and but still reliable method for sensitivity analysis [10]. The main idea is to take the expression for the goal function f into account and to derive new expressions that can be used to calculate for the sought terms. As an example, we consider minimum global elastic strain energy (compliance) as a goal function, i.e.

$$f(\mathbf{x}, \mathbf{u}) = \frac{1}{2} \mathbf{u}^{\mathrm{T}} \mathbf{f} \tag{2.10}$$

where f depends on both the design variables $\mathbf{x}$ and the displacement vector $\mathbf{u}$. The sensitivity analysis, i.e. the total derivative with respect to all x_i then can be written as

$$\frac{\mathrm{d} f(\mathbf{x}, \mathbf{u})}{\mathrm{d} x_i} = \frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial x_i} + \sum_j \left(\frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial u_j} \frac{\partial u_j}{\partial x_i} \right) \tag{2.11}$$

For the goal function in Eq. 2.10, the terms on the right-hand side of Eq. 2.11 are as follows:

$$\frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial x_i} = 0 \tag{2.12}$$

$$\frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial u_j} = \frac{1}{2} f_j \tag{2.13}$$

The remaining unknown term is $\partial u_j / \partial x_i$ which can be obtained by the *direct method* or *adjoint method*.

[2] Unless there are reasons to do otherwise, forward or backward differences are favoured over central differencing because the latter scheme requires the evaluating of f for one more set of x.

The *direct method* utilizes the static finite element equation and takes the derivative with respect to the design variables x_i:

$$\frac{\partial \mathbf{K}(\mathbf{x})}{\partial x_i}\mathbf{u}(\mathbf{x}) + \mathbf{K}(\mathbf{x})\frac{\partial \mathbf{u}(\mathbf{x})}{\partial x_i} = \frac{\partial \mathbf{f}}{\partial x_i} \tag{2.14}$$

The expressions are re-ordered then to give

$$\mathbf{K}(\mathbf{x})\frac{\partial \mathbf{u}(\mathbf{x})}{\partial x_i} = \frac{\partial \mathbf{f}}{\partial x_i} - \frac{\partial \mathbf{K}(\mathbf{x})}{\partial x_i}\mathbf{u}(\mathbf{x}) \tag{2.15}$$

$$\mathbf{K}(\mathbf{x})\frac{\partial \mathbf{u}(\mathbf{x})}{\partial x_i} = \phi(\mathbf{x}, \mathbf{u}) \tag{2.16}$$

It can be seen that the final form resembles the initial static finite element equation. However, instead of $\mathbf{u}(\mathbf{x})$ the sought for $\partial u_j/\partial x_i$ appears on the left-hand side and can be solved for. The right-hand side also contains $\partial \mathbf{K}(\mathbf{x})/\partial x_i$ that can be calculated analytically (analytical sensitivity analysis) or by use of finite differences (semi-analytical sensitivity analysis).
From Eq. 2.16 it can be seen that a total of n additional *pseudo load cases* have to be solved in order to determine $\partial u_j/\partial x_i$. Obviously, this is very costly for topology optimization where there is a large number of design variables. Thus, this method is only recommended for optimization problems with few design variables, otherwise the *adjoint method* is beneficial.

The starting point of the *adjoint method* is by looking at the total derivative of a constraint function:

$$\frac{\mathrm{d}g(\mathbf{x}, \mathbf{u})}{\mathrm{d}x_i} = \frac{\partial g(\mathbf{x}, \mathbf{u})}{\partial x_i} + \sum_j \left(\frac{\partial g(\mathbf{x}, \mathbf{u})}{\partial u_j}\frac{\partial u_j}{\partial x_i} \right) \tag{2.17}$$

From Eq. 2.16, the expression for $\partial \mathbf{u}(\mathbf{x})/\partial x_i$, i.e.

$$\frac{\partial \mathbf{u}(\mathbf{x})}{\partial x_i} = \mathbf{K}^{-1}(\mathbf{x})\left(\frac{\partial \mathbf{f}}{\partial x_i} - \frac{\partial \mathbf{K}(\mathbf{x})}{\partial x_i}\mathbf{u}(\mathbf{x}) \right) \tag{2.18}$$

is inserted in Eq. 2.17:

$$\frac{\mathrm{d}g(\mathbf{x}, \mathbf{u})}{\mathrm{d}x_i} = \frac{\partial g(\mathbf{x}, \mathbf{u})}{\partial x_i} + \sum_j \left(\frac{\partial g(\mathbf{x}, \mathbf{u})}{\partial u_j}\mathbf{K}^{-1}(\mathbf{x})\left(\frac{\partial \mathbf{f}}{\partial x_i} - \frac{\partial \mathbf{K}(\mathbf{x})}{\partial x_i}\mathbf{u}(\mathbf{x}) \right) \right) \tag{2.19}$$

After introducing a coefficient vector λ_j

$$\lambda_j = \left(\frac{\partial g(\mathbf{x}, \mathbf{u})}{\partial u_j}\mathbf{K}^{-1}(\mathbf{x}) \right)^{\mathrm{T}} = \mathbf{K}^{-1}(\mathbf{x})\left(\frac{\partial g(\mathbf{x}, \mathbf{u})}{\partial u_j} \right)^{\mathrm{T}} \tag{2.20}$$

this equation can be re-written in a way that it resembles the actual static finite element equation where $\mathbf{u}$ is replaced by $\boldsymbol{\lambda}$ and $\mathbf{f}$ by a *pseudo load case*:

$$\mathbf{K}(\mathbf{x})\boldsymbol{\lambda} = \left(\frac{\partial g(\mathbf{x}, \mathbf{u})}{\partial u_j} \right)^{\mathrm{T}} \tag{2.21}$$

After $\boldsymbol{\lambda}$ then has been solved for, it can be inserted in Eq. 2.19 again to calculate the sensitivity. Compared to the direct method, the total number of pseudo load cases to be solved is $m + p + 1$, i.e. one for each constraint equation and one for the goal function itself.
To conclude, it can be stated that if $m + p + 1 < n$, then the adjoint method is less costly than the direct method which is typical for topology optimization. Otherwise, the direct method should be favoured.

2.1.5 Approximation methods

The sensitivity analysis (Ch. 2.1.4) is very costly. In order to reduce computation time, the actual goal and constraint function are approximated by simpler functions. Two methods can be distinguished: *local approximations* and *global approximations*. It is noted already that only local methods are implemented in the finite element software PERMAS for topology optimization (Ch. 2.2.2.3) and shape optimization (Ch. 2.2.3.3).

2.1.5.1 Local approximation

Local approximation methods attempt to provide an approximation that is only valid in the vicinity of the current design variable vector $\mathbf{x}$ but not for the total design space. Therefore, these expressions are simpler and less costly but are also only valid for the current iteration. Once $\mathbf{x}$ has changed, a new local approximation has to be calculated.
Local approximations often include only linear terms of a Taylor series expansion as in Eq. 2.22 and prove good convergence at low computational cost. Other methods also include second order terms for better accuracy but at significantly increased cost, see Eq. 2.23.

$$f(\mathbf{x}) \approx f(\mathbf{x}^{(k)}) + \nabla^{\mathrm{T}} f(\mathbf{x}^{(k)})(\mathbf{x} - \mathbf{x}^{(k)}) \tag{2.22}$$

$$f(\mathbf{x}) \approx f(\mathbf{x}^{(k)}) + \nabla^{\mathrm{T}} f(\mathbf{x}^{(k)})(\mathbf{x} - \mathbf{x}^{(k)}) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^{(k)})\mathbf{H}(f)(\mathbf{x} - \mathbf{x}^{(k)}) \tag{2.23}$$

In above equations, $\mathbf{H}(f)$ is the *Hessian matrix*. Hereby, the coefficients of $\mathbf{H}(f)$ which are second-order derivatives of $f(\mathbf{x})$ with respect to $\mathbf{x}$ can be obtained at reduced cost by using the *BFGS algorithm* [11, p. 136] which is presumably the most popular quasi-Newton method.

A very early local approximation scheme is the so-called *SLP* (sequential linear programming) by *Fleury* in 1973 [10] that uses cheap linear approximation. Computationally more expensive is using quadratic approximation as in *SQP* (sequential quadratic programming) [11].

Very popular in commercial codes are non-linear programming schemes such as *CONLIN* (convex linearization [12, 13] also by *Fleury* in 1986 that is based on convex approximation. CONLIN was extended to a method called *MMA* (method of moving asymptotes) [14] in 1987 which was even further modified in 1989 by *Fleury* again [15]. Due to its good performace, MMA is the standard method for topology optimization in Altair OptiStruct.

Another very popular convex method is *SCP* (sequential quadratic programming) that was also developed by *Fleury* in 1993 [16]. A generalized convex approximation scheme is given in [17] for the so-called *GCA* (generalized convex approximation,1996). GCA has evolved to one of the most important methods in PERMAS besides SCP, SLP and SQP.

More approximation schemes are given in literature such as the *SRSM* (successive response surface method) [6, p. 85], *WLIN* or *LDR* and some of them have been implemented in PERMAS as well. However, their use is mainly restricted to special purposes.

2.1.5.2 Global approximation

Global approximation methods provide an approximation that is valid for the entire design space and therefore is not restricted to only one iteration but can be used for the entire optimization. Obviously, one must expect that a global approximation cannot be achieved with simple functions which is the advantage of local approximations. The following aspects have a significant influence on the quality of a global approximation.

- Size of the design space: the larger the design space, the more difficult it is to find a good global approximation. The experienced user should always consider a reasonable reduction of the design space when using global approximations.
- Choice of functions for approximation: a poor choice of functions might even not allow a satisfying approximation at all. The nature of the optimization problem should always be considered when choosing functions to describe it.
- Number and distribution of basic values: a large number of basic points for a high accuracy is costly. A good distribution also is beneficial for the quality of the approximation.

Basically, there are two types of global approximations: *smooth approximations* which do not include calculated basic points of f while *interpolations* do. The interested reader is referred to [6]. As mentioned before, no global approximation scheme is implemented in PERMAS.

2.1.6 Multi-objective optimization

An extension to the above is multi-objective optimization. This problem type involves not only one but several goal functions $f_i(\mathbf{x})$. The minimization problem in Eq. 2.1 can then be reformulated as

$$\begin{array}{lll} \text{minimize} & F\left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_i(\mathbf{x})\right) & , \quad \mathbf{x} \in \mathbb{R}^n,\ i = 1 \ldots l \\ \text{subject to} & g_j(\mathbf{x}) \leq 0 & , \quad j = 1, \ldots, m \\ & h_k(\mathbf{x}) = 0 & , \quad k = 1, \ldots, p \end{array} \tag{2.24}$$

with a new goal function F that depends on several $f_i(\mathbf{x})$. In the general case, there is no distinct optimum because the goal functions f_i are not minimized for the same solution $\mathbf{x}^*$. Instead, the goal is to find the so-called *Pareto optimality*. A Pareto optimal design is characterized by that fact that all its goal functions $f_i(\mathbf{x})$ are at least as good as for any other design and that there is at least one goal function that is better than for this other design [2]. Naming the Pareto optimal solution $\mathbf{x}^*$, this property can be stated as follows:

$$f_i(\mathbf{x}^*) \leq f_i(\mathbf{x}) \ , \qquad \forall i = 1, \ldots, l \tag{2.25}$$

$$f_i(\mathbf{x}^*) < f_i(\mathbf{x}) \ , \qquad \exists i \in \{1, \ldots, l\} \tag{2.26}$$

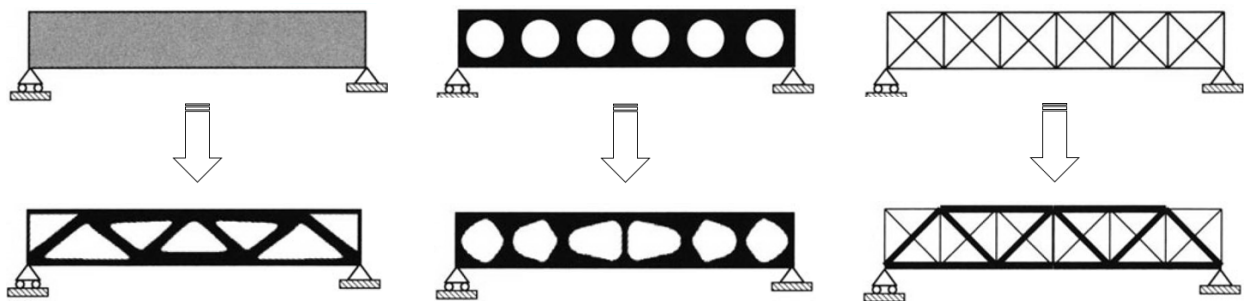
A straight-forward way to obtain a Pareto optimal solution is to introduce weight factors k_i and to take a linear combination of all weighted objective functions as a new goal:

$$F = \sum_{i=1}^{l} k_i f_i(\mathbf{x}) \qquad \text{with} \quad \sum_{i=1}^{l} k_i = 1 \tag{2.27}$$

It is recommended that the sum of all weight factors is k_i to enable a comparison of the value of f (single objective function) to the value of F (multiple objective function), if needed. There are further approaches on how F looks like, depending on the problem given.
An application of multiple-objective optimization from daily business is given for multiple load cases in structural finite element analysis. Hereby, each objective function f_i refers to one load case and a function F that weighs these functions is to be minimized, see Ch. 2.2.1.3.

2.1.7 Classes of structural optimization

Structural optimization can be divided into three classes: *topology optimization* as the most general form aims to find the distribution of material within a given design space in order to sustain a given load the best possible way (Fig. 2.1a). In this method, the maximum degree of freedom is given.
Shape optimization on the other hand only considers a modification of the outer surface, i.e. the boundary of the structure, while the basic distribution of material already has been decided (Fig. 2.1b). Thus, no new topologies like voids can be created.
Last, *sizing optimization* concerns the modification of thickness, cross-section or material parameters (Fig. 2.1c). These three classes of structural optimization are discussed in more detailed in the sequel.



(a) *Topology optimization* (b) *Shape optimization* (c) *Sizing optimization*

Figure 2.1: *Classes of structural optimization [3]*

2.1.7.1 Topology optimization

The goal of topology optimization is to find a distribution of material and voids over a user defined spatial domain, denoted *design space*. With regard to finite element modelling, this requires that the whole design space is meshed and the outcome of the iterative optimization is which elements are filled with material and which ones are "voids".
This distribution targets to be optimal in terms of a defined design goal function and possibly subject to certain design constraints. A common goal function in structural design is to maximize "stiffness" which can be achieved by minimizing the global elastic strain energy. A typical constraint is a upper weight limit that requires that only parts of the design space can be filled with material such that voids remain. Also, displacement or reaction force constraints could be defined if the type of problem requires this.

The three best-known methods for solving topology optimization problems numerically are *SIMP* (Solid Isotropic Material with Penalization), the *homogenization method* and *ESO* (Evolutionary Structural Optimization). SIMP and the homogenization method have been implemented in the finite element software PERMAS. SIMP has evolved as the standard method in most applications. However, the homogenization method is still selected by default in PERMAS mainly for historical reasons.
Both in SIMP and the homogenization method, the representation of finite elements that are either filled with material or empty is achieved by the introduction of the so-called *element filling ratio* x_i. This scalar can take up values between 0 and 1 and serves as a design variable for each finite element during the optimization process. A filling ratio of 0 represents a void, whereas a value of 1 represents material. Starting from initial fillings ratios x_i at the beginning of the optimization analysis (often $x_i = 0.5$), the element filling ratios are iteratively determined to be either (close to) 0 or to be (close to) 1 at the end of a converged optimization.
The procedure of ESO is slightly different as shown in the following paragraphs.

SIMP (Solid Isotropic Material with Penalization) Here, the element filling ratios x_i are directly related to material properties $\boldsymbol{\rho}$ and $\mathbf{E}$ of the finite elements. This approach is frequently denoted *Power Law* and was first considered by Bendsøe [18]. It is applicable for both isotropic and anisotropic material:

$$\boldsymbol{\rho}_i = x_i \, \boldsymbol{\rho}_0 \tag{2.28}$$

$$\mathbf{E}_i(x_i) = x_i^p \, \mathbf{E}_0 \quad , \quad p > 1 \tag{2.29}$$

where p is called the *penalty factor* and $\boldsymbol{\rho}_0$ and $\mathbf{E}_0$ are the actual element mass density and stiffness matrices for $x_i = 1$, respectively. Usually, $p = 3$ as recommended by PERMAS. For modal analyses, however, reducing the the penalty to factor ($p \approx 2$) should be considered to avoid undesired local modes in areas of low density [19, p. 1880].
The requirement $\varepsilon < x_i$ prevents the stiffness matrix $\mathbf{K}$ to become singular. Considering that $0 < \varepsilon < x_i < 1$ and $1 < p$, it can be seen that the element stiffness is penalized to a larger extend than the mass density. The motivation for a larger penalty factor is to achieve a better 0/1-distribution due to the worse relation between mass (cost) and stiffness (benefit) for intermediate fillings ratios.

Homogenization Method This method [18] proposes to consider the material as a microstructure that consists of an infinite number of small units cells. These periodic unit cells have internal voids that can be shaped as square or rectangular holes or are defined as layered structures. Therefore, the microstructure is orthotropic, in general. The size of the voids in relation to the size of the unit cell defines the filling ratio μ. This method then tries to relates the filling ratio μ to the Young's modulus E and to derive a homogenized material property that describes its macroscopic behaviour but depends on the microscopic parameters of the unit cell. For a rectangular void, these parameters are the normalized length a of a side and the angular orientation θ:

$$\alpha = a^2 \tag{2.30}$$

$$\mu = 1 - \alpha \tag{2.31}$$

$$E_{ijkl} = E_{ijkl}\,(\alpha, \theta) \tag{2.32}$$

where α is the hole size and E_{ijkl} the elasticity tensor of the microstructure. When the goal function is stiffness,

a weight constraint can be introduced via an upper volume bound:

$$\int_{\Omega} \mu\left(\mathbf{x}\right) \mathrm{d}\Omega \leq V_{\max} \tag{2.33}$$

The drawback of the homogenization method is that each element is characterized by not only one but all variables needed to describe the void which can be costly. In the most general three-dimensional case, there are five design variables: three for the edge lengths of the hole and two angles for the orientation in the unit cube.

ESO (Evolutionary Structural Optimization) This method [20] is based on the idea that by slowly removing inefficient material from a given structure, the optimal shape will be found. Inefficient material is identified as it is subjected to low stresses and therefore is under-utilized. When this material is removed subsequently, a more uniform stress distribution arises which represents good material utilization at an optimized weight.
In practice, a linear-static stress analysis is performed for the initial design space first. The stress state of each element can be measured by an equivalent stress, e.g. the von Mises equivalent stress σ_{vM}. All under-stressed elements which satisfy the relation

$$\frac{\sigma_{\mathrm{vM}}}{\sigma_{\mathrm{vM,max}}} < RR_i \tag{2.34}$$

are completely deleted from the model. Here, RR_i is the user-defined, current rejection ratio. A typical value for RR_0 is 1 %. This procedure is repeated until no elements satisfy the relation 2.34 anymore which is called the *steady state*. Afterwards, an *evolutionary rate* (ER) (also typically 1 %) is added to the initial rejection ratio as

$$RR_{i+1} = RR_i + ER \tag{2.35}$$

and the procedure is continued until the steady state is reached again. A convergence criterion can be defined such that the minimum stress level is not less than a certain percentage less then the maximum stress. It should be noted that only in special cases, a complete uniform stress distribution can be achieved.
The ESO also allows to define stiffness and displacement constraints. However, these two quantities cannot be taken as goal function of the optimization which due to the nature of ESO.

2.1.7.2 Shape optimization

The subject of shape optimization is the modification of the boundary of a structure according to a goal function and optional constraints. Unlike topology optimization, new boundary (e.g. holes) cannot be created. In daily practice, there are two types of this method: *CAD-based shape optimization* and *FEM-based shape optimization*.

CAD-based shape optimization The first method is based on a parametrized CAD-model that can be accessed by the optimization solver. During the optimization, the parameters are varied within defined limits and the CAD-model is updated accordingly. Each shape variation is meshed afterwards and proper boundary conditions and loads are applied. There exist CAD programmes like UNIGRAPHICS or Pro/Engineer where the whole process of CAD-model modification and automatic meshing has been implemented [6, p. 156].
Unless the shape optimization is restricted to the modification of simple details like the radius of a fillet, it is not always a simple task to create a CAD model that is suitable for shape optimization. Furthermore, many shape optimization solvers cannot access CAD files or do not have a suitable interface for a CAD programme why the following method is very popular.

FEM-based shape optimization Here, a shape optimization can be performed based on a finite element mesh that is independent from the basic CAD model. Again, two methods can be distinguished: the mathematical optimization algorithm that utilizes shape basis vectors for shape modification and CAO (Computer-Aided-Optimization) which is based on optimality criteria.

Shape optimization using shape basis vectors *Shape basis vectors* provide a normed displacement value for each degree of freedom inside the design space and therefore specify how the design space and most importantly its boundary deform when this shape basis vector is scaled. Thus, the shape basis vector can be seen as a displacement field $\mathbf{u}$ that is multiplied by a scale factor k and then added to the initial nodal coordinates $\mathbf{x}^0$ of the design space:

$$\mathbf{x} = \mathbf{x}^0 + k\,\mathbf{u} \tag{2.36}$$

Hence, the shape modification completely depends on the *shape basis vector* which is usually calculated by the optimization solver because it involves complex algorithms. For more details, see [6, p. 160f.].
The user's input for the calculation of a shape basis vector is to prescribe how the shape modification should look like. This is conveniently done by selecting some design nodes on the surface of the design space that is to be modified, e.g. four nodes on a radius fillet. Every design node is assigned a design variable which specifies in which direction the node should be displaced during the optimization (e.g. a surface-normal displacement). This "degree of freedom" is furthermore bounded by some user-specified modification limits. Based on the prescribed displacement directions of the design nodes, the optimization solver then smoothly interpolates the displacement directions of all remaining nodes inside the design space which finally gives the shape basis vector.
If there are many design nodes, it can happen that the shape basis vector does not lead to smooth shape modifications and undulating surfaces exist. Then, it is recommended to create one shape basis vector for each design node. The solver then computes individual displacement fields $\mathbf{u}_i$ and finally superposes them such that:

$$\mathbf{x} = \mathbf{x}^0 + \sum_{i=1}^{n} (k_i \mathbf{u}_i) \tag{2.37}$$

where n is the number of design variables. This procedure reliably leads to smooth shape modifications that can be adopted in structural design.
To conclude, only a handful of design variables are defined for a shape optimization whereas in topology optimization each finite element is assigned one. This has a major effect on design sensitivity analysis.

2.1.7.3 Sizing optimization

This type of structural optimization is not applied within this project and therefore is only addressed briefly. Sizing optimization concerns solely the variation of (scalar) parameters such as cross-section dimensions, thickness or material parameters.
An interesting area of application are composite materials where the number of layered plys, their various thicknesses and orientations can be optimized. Another typical problem is the positioning of struts in wall-like structures that can be optimized aiming at stress reduction or elevated eigenfrequencies.

2.1.8 Problems in numerical structural optimization

2.1.8.1 Local vs. global optimum

Most structural optimization solvers use local approximation methods by default. The solution obtained from these methods are, in general, only local optimum but not necessarily the global one. Especially models with various types of constraints can become highly non-convex which may lead to the existence of a large umber of local minima [21]. Therefore, results should always be assessed with a critical view.

2.1.8.2 Checkerboarding and mesh dependency

Frequently encountered problems in topology optimizations are *checkerboard*-like material distributions that are caused by errors in the finite element formulation. Closely related to the problem of mesh dependency is the minimum member size control[3]. For decreasing element size, an increased number of smaller members is found. These numerical problems have been assessed in various papers such as [22] and have been successfully implemented in modern optimization codes.

[3]Member size is the size of coherent mass of a structure. In topology optimization this refers to the size of connected (and filled) finite elements.

2.2 Applied structural optimization with finite element software

The following section treats fundamental aspects about performing structural optimizations with finite element software. First, the general procedure of setting up a finite element model is covered. A side note on the use of consistent units in FEM is given in App. A. Second, parameters of topology and shape optimization and their influence are described, in particular.

2.2.1 General aspects about finite element modelling

2.2.1.1 Finite elements and mesh size

Modelling a loaded structure with finite elements, simplified boundary conditions and loads is always only an approximation of reality. Thus, a thoughtful strategy can only reduce the error that comes along with this method [23].
The finite element software PERMAS offers a wide range of element types, see *Element Atlas* [24, p. 1414]. However, the choice of element types boils down to three-dimensional solid (continuum) interface elements since all investigated engine components for structural optimization show significant spatial extent in all three dimensions. The degrees of freedom for displacement (PERMAS DOF-Type `DISP`) are translation (u, v, w) at each node.
Standard solid interface elements are tetrahedra, pentahedra (wedges), pyramids and hexahedra (bricks). In general, tetrahedra can mesh any geometry and many automatic meshing algorithms[4] are available which makes the use of this element type very convenient. However, more accurate results are obtained with brick elements which should always be favoured [25]. Therefore, tetrahedra, wedges or pyramids should only be used when brick meshing is too cumbersome [24, p. 1488]. Based on this, the investigation of element types was restricted to the most common brick and tetrahedral elements (Tab. 3.1).

Geometry	**Order of shape functions**	**Number of nodes**	**PERMAS name**
Brick elements	linear	8	`HEXE8`
	(incomplete) quadratic	20	`HEXE20`
Tetrahedral elements	linear	4	`TET4`
	quadratic	10	`TET10`

Table 2.1: Considered solid elements in PERMAS

Hereby, elements can be formulated with linear, incomplete quadratic and (fully) quadratic shape functions. Linear elements are defined only by their corner nodes and therefore heave straight edges. Incomplete quadratic elements (serendipity elements) also include midside nodes and have straight or curved (parabolic) edges. (Fully) quadratic elements (Lagrangian elements) furthermore include midface nodes. The latter elements are computationally very expensive and therefore are seldom used for structural analysis at Porsche. Thus, they are not considered any further, leaving only four elements for comparison (Figure 3.1).
The goal of this paper is to determine the dependence of the results from the choice of finite elements and their discretization. Regarding computational effort, it is always favoured to choose the coarsest mesh with 'cheap' linear elements that give sufficiently exact results.

2.2.1.2 Boundary conditions

The representation of boundary conditions has to be considered properly because of the influence on the force flux inside the structure. Clearly, this has a major effect on the calculated displacement field (u, v, w) or the element stress tensor σ_{ij} which are frequently taken as goal or constraint functions for topology and shape optimization, respectively.
In many cases, components of interest are analysed without their connected counterparts. This could be because they do not exist yet or to keep the model small. However, clamped or simply supported boundaries are a rough simplification and do not consider the actual elastic behaviour of the whole compound structure. Therefore, the boundary conditions can be overly stiff or even allow mode shapes which do not occur in the

[4] Automatic tetrahedral meshing is available in the preprocessors ANSA, MEDINA Pre and SimLab which are used at Porsche.

compound structure. Particularly the bad stress prediction in these cases can lead to difficulties if stress is used as a goal or constraint function which is typical for shape optimization. Hence, extended models that also include connected components are recommended.

2.2.1.3 Multiple load cases

A common requirement in finite element analysis is the consideration of different load cases. Each load case represents an idealized loading situation and they can differ largely in magnitude and direction. As an example, we consider an engine component under acceleration during a turning manoeuvre. Hereby, the components underlies both longitudinal and lateral acceleration.
Regarding the engine mount bracket that is analysed in this paper, there exist three different types of loadings which are sorted according to magnitude in Tab. 2.2.

Loading type	**Material response**	**Examples**
Operating load	Only elastic, operational strength required	Acceleration, braking, cornering
Misuse loadings	Only elastic	Heavily driving over kerb, 'soft' collision
Crash loadings	Plastic response allowed but no fracture	Legal requirements for components

Table 2.2: Loading types

A thoughtful strategy for considering these loads is inevitable. First, it is necessary to assess which load cases are relevant for the optimization problem in question. If more than one load case remains (multiple load case), the weighting has to be defined since it significantly influences the results of any optimization. Also, load combination like combined braking and cornering have to be considered.
In topology optimization, the goal is to find a structure that is maximal stiff with respect to the given loads that appear in the force vector $\mathbf{f}$. Maximizing stiffness is equivalent to minimizing the global elastic strain energy:

$$\text{minimize} \qquad f(\mathbf{x}) = C(\mathbf{x}) = \frac{1}{2}\mathbf{u}^{\mathrm{T}}(\mathbf{x})\mathbf{K}(\mathbf{x})\mathbf{u}(\mathbf{x}) = \frac{1}{2}\mathbf{u}^{\mathrm{T}}(\mathbf{x})\mathbf{f} \tag{2.38}$$

As can be seen in Eq. 2.38, linear scaling of the force vector $\mathbf{f}$ might change the value of $f(\mathbf{x})$ but it does not change the solution $\mathbf{x}$ of the minimizing problem. It is solely the weight of the single components of $\mathbf{f}$ that has an effect on the solution. Therefore, the absolute magnitude of forces is not of importance, it is only their relative magnitude that counts.
Two different weighting strategies for multiple loads are considered in the sequel. The first approach is based on the idea that solely the load pattern with the largest goal function f_i is considered and optimized while all other load patterns are ignored completely. Thus, the minimization problem can be stated as follows:

$$\begin{aligned} \text{minimize} \qquad F(\mathbf{x}) &= \max\left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_n(\mathbf{x})\right) \\ &= \max\left(C_1(\mathbf{x}), C_2(\mathbf{x}), \ldots, C_n(\mathbf{x})\right) \end{aligned} \tag{2.39}$$

where f_i are the scalar goal function values of n different load cases. If no settings are changed, this is the default method in PERMAS. The huge drawback hereby is the fact that only one load case is optimized in each iteration. In the worst case, the goal function of only one load pattern always remains larger than all other goal functions so that the latter ones are not considered at all. The resulting structure is then only optimal for one load case but not for the remaining ones.
A more conservative approach that bypasses the risk of neglecting "small" load cases is to use weight factors k_i. These factors can be chosen from experience (constant weight factors) or by means of a mathematical expression that is evaluated by the solver and possibly updated every iteration (dynamic weight factors). The optimization problem can then be reformulated:

$$\begin{aligned} \text{minimize} \qquad F(\mathbf{x}) &= k_1 f_1(\mathbf{x}) + k_2 f_2(\mathbf{x}) + \ldots + k_n f_n(\mathbf{x}) \\ &= k_1 C_1(\mathbf{x}) + k_2 C_2(\mathbf{x}) + \ldots + k_n C_n(\mathbf{x}) \end{aligned} \tag{2.40}$$

where k_i are the real and positive weight factors. For convenience, the sum of all weight factors is chosen to be

equal to 1 which allows for a comparison of a multiple load case and a single load case goal function:

$$\sum_{i=1}^{n} k_i = 1 \tag{2.41}$$

A common approach for dynamic weight factors is to assure that small load cases are considered to a higher extend than their rather small goal function f_i would suggest. The motivation behind this is that all load cases are treated equally and thus, the structural design can sustain all loads equally well. This is achieved by equating all weighted goal functions in the following fashion:

$$k_1 C_1 = k_2 C_2 = \ldots = k_n C_n = \text{const.} \tag{2.42}$$

Combined with Eq. 2.41, the unknown (and dynamic) weight factors in Eq. 2.40 can be substituted by expressions that only depend on the known goal function values f_i.
For this non-standard goal function, PERMAS is not capable of performing a gradient-based sensitivity analysis, i.e. direct or adjoint method. Instead, the sensitivity analysis must be done using finite differences and therefore requires explicit expressions for the first and second partial derivatives of the goal function. All this input must be provided with a Fortran subroutine that is called by the optimization solver. The implementation is described in the method chapter, the code is found in App. C.1.3.

2.2.1.4 Material models

The optimization results are furthermore determined by the choice of material model. These models are defined by constitutive equations that relate *primary* field variables like the displacement vector $\mathbf{u}$ to *secondary* field variables like the stress tensor $\boldsymbol{\sigma}$.
A material is called *ideally* or *Cauchy* elastic when, under isothermal conditions, a body recovers its initial shape completely after removal of loads causing its deformation. Then, there exists a linear relationship between the stress tensor $\boldsymbol{\sigma}$ and the strain tensor $\boldsymbol{\varepsilon}(\nabla \mathbf{u})$ which is function of the gradient of the primary displacement field $\mathbf{u}$.
The stress-strain relationship for a linear elastic solid is described by the so-called generalized Hooke's law:

$$\sigma_{ij} = C_{ijkl}\, \varepsilon_{kl} \tag{2.43}$$

where C_{ijkl} is the general fourth-order stiffness tensor. Due to symmetry conditions, the number of independent coefficients is reduced to 21 [26, p. 181]. If furthermore isotropy, the independence of material behaviour of direction, is assumed, the total number of independent material parameters in C_{ijkl} is reduced to 2, namely the Young's modulus E and the Poisson's ratio ν.

The materials used for the engine components in this paper are conventional metals. For these homogeneous materials, it is a fair assumption to consider them to be linear elastic and isotropic when subjected to stresses below the yield limit [25]. In addition, no temperature dependence is assumed because the temperatures of the investigated components do not reach values where a noticeably different material behaviour compared to room temperature is expected.
It should also be mentioned that the generalized Hooke's law is only valid for small strains. In fact, metals can only carry small amount of elastic strain which can be seen by comparing the Young's modulus E to the yield stress σ_y in the uni-axial stress test. The required material parameters are listed in Tab. 2.3.

Material parameter	**Symbol**	**Unit used for FEM**
Young's modulus	E	$[\mathrm{N} \cdot \mathrm{mm}^{-2}]$
Poisson's ratio	ν	[-]

Table 2.3: Material parameters for isotropic, linear elasticity

Constant values for E and ν are taken from experiments that were conducted on the corresponding materials. These experiments were performed at room temperature $\vartheta = 20\,^\circ$ and are assumed to vary little at operating temperatures of the analysed engine components.

2.2.2 Fundamentals of numerical topology optimization

2.2.2.1 Design space

The starting point of each topology optimization is given by providing a suitable design space. It constitutes the available domain in which the solution is found, requiring that the defined problem is solvable in the first place. In practice, the design space is provided by the design department that sets up a CAD model of the entire vehicle.

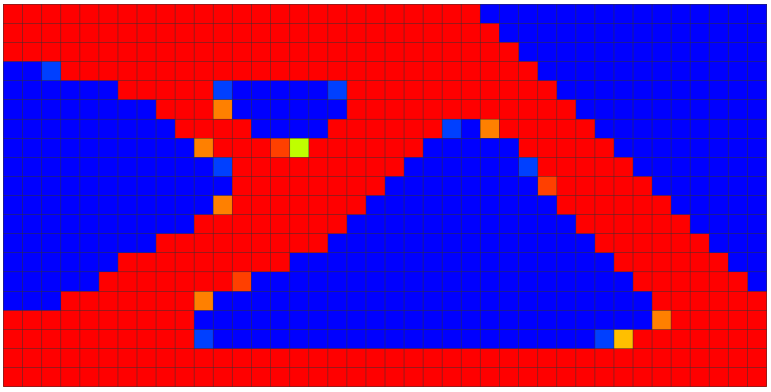


Figure 2.2: *Topology optimization result inside a cuboid design space*

It should be noted that the size of the design space directly influences the number of finite elements used for meshing and therefore increases the computational cost. A large design space might require to coarsen the mesh, possibly leading to inaccurate results. Thus, it is up to the user's experience to remove parts of the actual design space that will certainly not contribute to the structural stiffness and to choose an element size that reliably represent structural features like a minimal wall thickness.

2.2.2.2 Goal and constraint functions

Goal and constraint functions can be any structural response that is calculated in a preceding analysis: model volume or weight, compliance, eigenfrequencies, element or nodal stresses and displacements.
A common goal function for topology optimization is minimum compliance C (global elastic strain energy) for a given load with a mass constraint, i.e. an upper weight limit $m_{\max}$. This results a maximum stiff structure:

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) = C(\mathbf{x}) = \frac{1}{2}\mathbf{u}^{\mathrm{T}}(\mathbf{x})\mathbf{f} = \frac{1}{2}\mathbf{u}^{\mathrm{T}}(\mathbf{x})\mathbf{K}(\mathbf{x})\mathbf{u}(\mathbf{x}) && , \quad \mathbf{x} \in \mathbb{R}^{n_{\mathrm{el}}} \\ &\text{subject to} && g(\mathbf{x}) = m(\mathbf{x}) = \sum_{j=1}^{n_{\mathrm{el}}} (\rho_j(\mathbf{x}) V_j) \leq m_{\max} && , \quad j = 1, \ldots, n_{\mathrm{el}} \end{aligned} \tag{2.44}$$

The displacement and force vector are denoted $\mathbf{u}(\mathbf{x})$ and $\mathbf{f}$, respectively, while $\mathbf{K}(\mathbf{x})$ is the stiffness matrix. Furthermore, $\rho_j(\mathbf{x})$ and V_j are the element mass density and volume. The design vector is of size $1 \times n_{\mathrm{ndof}}$ where n_{ndof} is the number of degrees of freedom inside the design space.
Stress and eigenfrequency as a goal or constraint function can cause numerical problems due to the nature of intermediate filling ratios combined with the penalty factor of the element stiffness matrix, see Ch. 2.1.7.1.

2.2.2.3 Sensitivity analysis and approximation methods

The method for sensitivity analysis (for more detail, see Ch. 2.1.4) in topology optimization is automatically chosen to be the adjoint method. Since topology optimization involves many design variables and few constraints, the adjoint method is advantageous over the direct method and certainly less costly than finite differences.
A number of different approximation methods (Ch. 2.1.5) has been implemented into PERMAS. Those are GCA, SCP, SLP and SQP as well as modified methods for sparse constraints (SCPS, SLPS, SQPS).

2.2.3 Fundamentals of numerical shape optimization

The following sub-sections are restricted to FEM-based shape optimization because only this type was performed within the present work.

2.2.3.1 Design space

The design space for shape optimization is limited to that part of the component where a modification of the boundary is intended. Considering that the mesh is being deformed during the optimization, several element layers under the surface should be included in the design space to avoid a heavily distorted mesh. Bad quality elements can cause poor analysis results, e.g. stresses, that are used as goal function or constraints. In the worst case, the analysis fails because the solver exits the computation. To avoid problems, all elements of the design space should be of good quality.

2.2.3.2 Goal and constraint functions

As in topology optimization, any response can be used as a goal function or constraint, i.e. model volume or weight, compliance, eigenfrequencies, element or nodal stresses and displacements.
However, a shape optimization usually aims to eliminate a structural response that could cause failure. In the usual case of critical stresses, there a two basic approaches to reduce stress.
The first and obvious method is to choose the minimization of element stress as the goal function such as

$$\text{minimize} \qquad f(\mathbf{x}) = \sigma_i(\mathbf{x}) \quad , \quad \mathbf{x} \in \mathbb{R}^{n_{\text{el}}} \tag{2.45}$$

This, by definition, results in the lowest possible stress state but since this optimization problem is not constrained, the component might be modified much more than needed to prevent failure. For example, if this is applied on a transition radius, the radius will be constantly enlarged until the design variables reach their user-defined limits. This means that the shape modification - in principle - is governed by the design variable limit which is certainly not intended. Furthermore, structural weight is increased simultaneously which is not desired either.
Given these reason, it seems more reasonable to use element stresses as a constraint rather than a goal function. Hence, an upper stress bound $\sigma_{\max}$ is defined by the user. The required goal function is defined as minimum weight for logical reasons. Otherwise, an undesired thickening of the structure might be the optimization result although the stress limit could be achieved with less material. Still, it should be noted that the weight of the optimized design space might be even higher than its initial weight although minimized mass has been specified as goal function. This seemingly contradiction can be explained by the fact that there is a stress violation initially that can only be eliminated by adding mass. It should be remembered that constraint violation has priority over goal function minimization. Formally, this problem is stated as follows:

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) = m(\mathbf{x}) = \sum_{i=1}^{n_{\text{el}}} (\rho_j V_j(\mathbf{x})) && , && \mathbf{x} \in \mathbb{R}^{n_{\text{el}}} \\ &\text{subject to} && g_j(\mathbf{x}) = \sigma_j(\mathbf{x}) \leq \sigma_{\max} && , && j = 1, \ldots, n_{\text{el}} \end{aligned} \tag{2.46}$$

In PERMAS, σ_j could be chosen as an equivalent stresses (e.g. von Mises equivalent stress σ_{vM}), a principal stress or a selected stress component from the nodal or element stress tensor.

In anticipation of Ch. 2.2.4.1, it is noted that multi-axial fatigue analyses outside of PERMAS are performed for engine components where an equivalent mean stress σ_m and amplitude stress σ_a is obtained for the given load history in FEM. The hereby calculated amplitude stress is significant for operational strength in the later problems. Thus, it is aimed at using σ_a for the stress constraint in Eq. 2.46.
However, amplitude stresses cannot be calculated in PERMAS directly but only by use of the software FEMFAT. Since an interface between these two programmes does not exist, a way of estimating amplitude stresses with PERMAS has to be developed:
The mean stresses σ_m in the multi-axial fatigue analysis come from constant loads which are always present in an engine, e.g. assembly loads such as bolt pretension. In contrast, amplitude stresses arise from external loads, e.g. cornering or acceleration, that appear repeatedly. Therefore, a time history is defined in FEM (Tab. 3.5) with time steps where only constant assembly loads are present and time steps where both assembly

and external loads exist. The estimation of the amplitude stress is then obtained by subtracting the stress state of the former time step of the latter one. This so-called *result combination* can conveniently be done by PERMAS and be used as the stress constraint in Eq. 2.46 as intended.

2.2.3.3 Sensitivity analysis and approximation methods

The sensitivity analysis (see Ch. 2.1.4) in shape optimization can be performed using finite differences or the (semi-)analytical direct and adjoint method. By default, the solver decides for its own which method is chosen. As usual for shape optimization, the number of constraints is larger than the number of design variables and therefore, the solver selects the direct method. Otherwise, the adjoint method is used. In some special cases, e.g. when a combination of non-linear results is performed during the optimization, no gradient-based methods (direct/adjoint method) can be used and therefore the more costly finite differences have to be selected.
The number of available approximation methods (Ch. 2.1.5) is very large: GCA, SCP, SLP, SQP, WLIN and LDR. The user is able to set the method to AUTO for which the solver automatically decides which method is best for the given problem.

2.2.4 Related analyses

A throughout component design requires further analysis types which protect against failure. Within this Master's thesis, the following analyses were performed apart from topology and shape optimization:

Software	**Analysis type**
FEMFAT	Multi-axial fatigue analysis (based on PERMAS linear-static analysis)
PERMAS	Linear-static analysis Modal analysis Linear buckling analysis

Table 2.4: Related analysis types

2.2.4.1 Multi-axial fatigue analysis

In modern vehicle development, a component generally does not only underlie static uni-axial loads that can be evaluated directly by a finite element calculation. Instead, components also have to sustain multi-axial and cyclic loads that can cause failure for mean and amplitude stresses much lower than the uni-axial yield limit. Thus, the fatigue safety factor calculation is one of the major tasks.
The basis for this fatigue analysis is the so-called Haigh diagram that shows values of mean and amplitude stresses that achieve infinite life[5]. This diagram can be derived from experiments or approximated by material parameters for ultimate, yield and fatigue strength.
The value of the safety factor SF is obtained by relating the point (σ_m, σ_a) of a stress cycle to an intersecting point of the infinite life line in the Haigh diagram. The method of constant stress ratio proposes that the intersecting point is found on the line that goes through the origin and (σ_m, σ_a). The method of constant mean assumes that the intersecting point lies on the vertical line where the mean stress σ_m remains constant. In this paper, the latter method is chosen since the change of the point (σ_m, σ_a) during shape optimization is expected to mainly influence σ_a.
Thus, the safety factor SF_A is defined as the ratio of maximum amplitude stress to the present amplitude stress and should globally be larger than a specified minimum safety factor for an approved design of a component.

$$SF_A = \frac{\sigma_{a,\max}(\sigma_m)}{\sigma_a(\sigma_m)} \geq SF_{A,\min} \tag{2.47}$$

To enable this calculation, the mean and amplitude stresses for operational loads have to be found. Here, the multi-axial fatigue analysis software FEMFAT is used. Input data are given as a sequence of (3D) element stress states from a linear static FEM analysis, see also Fig. C.8. These element stress states correspond to different (constant amplitude) operating loads in the load history.

[5]Diagrams can also be derived for achieving N load cycles. However, infinite life is of main interest here.

The stresses that are used to calculate σ_m and σ_a are selected from critical plane with the highest normal stress which is considered to cause maximum damage [27]. The critical plane criterion has to be considered for multi-axial fatigue since the directions of principal stresses are changing permanently and therefore, classic criteria (maximum principal stress, von Mises stress, etc.) are not applicable. In order to identify the critical plane, several steps are necessary. First, all load tensors σ_{ij} are transformed into several planes and the most interesting ones are filtered. For these planes, load histories of stress components are generated. Next, rainflow counting and a damage analysis are performed. The cutting plane that shows the maximum normal stress is then assumed to be critical for the fatigue life. For all time steps, the maximum and minimum stresses in the time history of these cutting planes are found and the worst combination is finally used to determine the mean stress σ_m and amplitude stress σ_a that are limited by the infinite life line in the Haigh diagram.
The FEMFAT module PLAST also considers local plasticizing after calculating stresses with the linear stress-strain curve. Both the mean and amplitude stresses are then redistributed on the cyclic stabilized stress-strain curve using the Neuber hyperbola.
The output of the fatigue analysis are values for damage, safety factors and damage factors. The interested reader is referred to [28, 27] for further information on the theory of multi-axial fatigue analysis with FEMFAT.

2.2.4.2 Modal analysis

In modal analysis, the natural mode shapes and corresponding eigenfrequencies during free vibration are determined. Using FEM, assuming linear elastic material behaviour and Hooke's law, the generalized equation of motion for the discrete system is given as

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{f}(t) \tag{2.48}$$

where $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ are the mass, damping and stiffness matrix, respectively. The displacement vector $\mathbf{u}$ also appears as velocity $\dot{\mathbf{u}}$ and acceleration $\ddot{\mathbf{u}}$ which are the first and second time derivatives.
For vibration analysis, the damping matrix $\mathbf{C}$ is usually omitted such that the problem reduces to

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{0} \tag{2.49}$$

During free vibration, it is assumed that the displacement $\mathbf{u}(t)$ and so its time derivatives are harmonic with respect to time:

$$\mathbf{u}(t) = \mathbf{u}_0 \cdot \mathrm{e}^{-i\omega_0 t} \tag{2.50}$$

$$\ddot{\mathbf{u}}(t) = -\omega_0^2 \cdot \mathbf{u}_0 \cdot \mathrm{e}^{-i\omega_0 t} = -\omega^2 \cdot \mathbf{u}(t) \tag{2.51}$$

where i is the imaginary unit with $i^2 = -1$ and ω_0 is the *eigenfrequency* or *natural frequency* of the system. Inserted in Eq. 2.49, the characteristic eigenproblem remains

$$\begin{aligned} -\omega_0^2 \cdot \mathbf{M}\mathbf{u}(t) + \mathbf{K}\mathbf{u}(t) &= \mathbf{0} \\ \left(-\omega_0^2 \cdot \mathbf{M} + \mathbf{K}\right)\mathbf{u}(t) &= \mathbf{0} \\ \rightarrow \qquad \det\left|-\omega_0^2 \cdot \mathbf{M} + \mathbf{K}\right| &= \mathbf{0} \end{aligned} \tag{2.52}$$

The non-trivial solution then gives the sought eigenfrequencies of the system.

2.2.4.3 Linear buckling analysis

Linear (eigenvalue) buckling analysis is used to estimate the critical (bifurcation) load of a "stiff" structure. A stiff structure is primarily subjected to axial loading that usually involves little deformation prior to buckling where the structure suddenly exhibits large displacements.
In a linear eigenvalue buckling analysis, the load for which the stiffness matrix becomes singular is sought for. Then,

$$\begin{aligned} \mathbf{K}\mathbf{u} &= \mathbf{0} \\ \left(\mathbf{K}_0 + \boldsymbol{\lambda}\mathbf{K}_\Delta\right)\mathbf{u} &= \mathbf{0} \\ \rightarrow \qquad \det\left|\mathbf{K}_0 + \boldsymbol{\lambda}\mathbf{K}_\Delta\right| &= \mathbf{0} \end{aligned} \tag{2.53}$$

has solutions other than the non-trivial solution $\mathbf{u} = \mathbf{0}$. The stiffness matrix $\mathbf{K}$ consists of a contribution of the base state stiffness matrix $\mathbf{K}_0$ and the contribution of the load stiffness matrix $\mathbf{K}_\Delta$ due to the incremental load q. From the last equation, the eigenvalues $\boldsymbol{\lambda}$ can be determined and thereafter the buckling mode shapes $\mathbf{u}$.

3 Method

This chapter deals with the methodology used within this project. The first part describes how the influence of settings and parameters for structural optimization was investigated by means of simple case studies. The obtained knowledge is then used for subsequent analyses on a real engine component. Particular attentions is devoted to the application of the used finite element and optimization software PERMAS and Optistruct.

3.1 Investigation of settings for structural optimization with FE software

In order to understand how to perform a structural optimization and which influence can be attributed to certain settings or parameters, simple academic examples are performed. Regarding topology optimization, an optimal distribution of a restricted mass inside a rectangular design space is obtained for a given boundary load. The shape optimization focuses on the modification of a transition radius subject to constraint stresses that are exceeded initially. Both are typical examples of structural optimization and furthermore utilize the same goal and constraint functions as the later engine mount bracket optimization.
The outcome of these basic problems is formulated into guidelines that are applied in real analyses on engine components later on. However, this does not only describe the set-up of the optimization situation itself but it also includes all preceding steps of finite element modelling.

3.1.1 Mesh size and element types

The finite element software PERMAS offers a wide range of element types, see *Element Atlas* [24]. However, the choice of element types is restricted to three-dimensional solid elements since all investigated engine components for topology or shape optimization show significant spatial extent in all dimensions. The most common solid elements for structural analysis are linear and (incomplete) quadratic bricks and tetrahedrons, see Tab. 3.1 and Fig. 3.1. They are used to mesh the design spaces in Ch. 3.1.3 and 3.1.4.

Geometry	Order of shape functions	Number of nodes	PERMAS name
Brick elements	linear	8	`HEXE8`
	(incomplete) quadratic	20	`HEXE20`
Tetrahedron elements	linear	4	`TET4`
	quadratic	10	`TET10`

Table 3.1: Solid elements available in PERMAS

There also exists a variety of special elements like wedges and pyramids or the computationally expensive fully quadratic brick elements but those are not considered here.

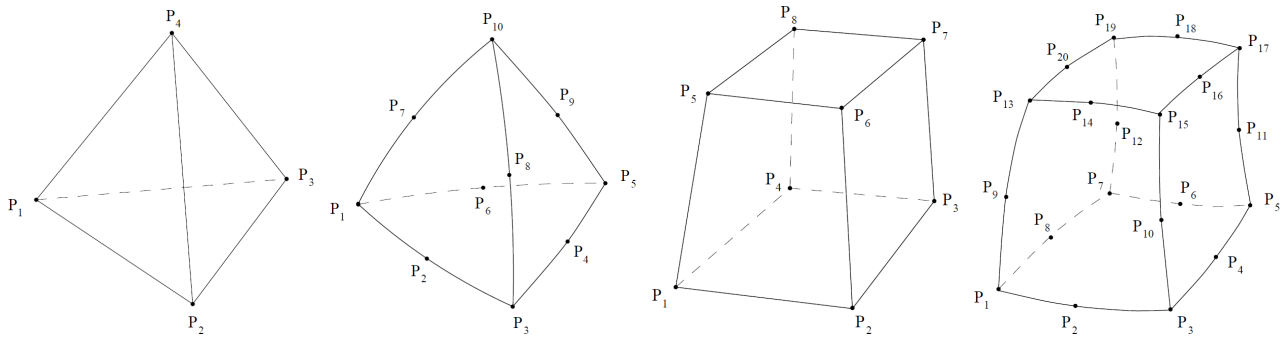


Figure 3.1: *PERMAS* `TET4`*,* `TET10`*,* `HEXE8` *and* `HEXE20` *elements [24]*

3.1.2 Material properties

The parameters for an isotropic material in a linear-elastic analysis are the Young's modulus E and the Poisson's ratio ν. To find out if the topology optimization result depends on these quantities, an identical optimization calculation is performed for typical steel ($E = 210\,\text{GPa}$, $\nu = 0.30$) and aluminium alloy ($E = 70\,\text{GPa}$, $\nu = 0.27$) and the results are compared. The input deck is specified as follows:

```
$ENTER MATERIAL
   $MATERIAL NAME = STEEL TYPE = ISO
      $ELASTIC GENERAL
         2.100000E+05 3.000000E-01
      $DENSITY GENERAL
         7.850000E-09
   $END MATERIAL
   $MATERIAL NAME = ALU TYPE = ISO
      $ELASTIC GENERAL
         7.000000E+04 2.700000E-01
      $DENSITY GENERAL
         2.700000E-09
        $END MATERIAL
$EXIT MATERIAL
```

3.1.3 Topology optimization

The exemplary problem for a topology optimization concerns seeking an optimal distribution of a limited amount of material within a restricted volume, see Fig. 3.2. Here, it was chosen that 50 % of the given design space should be filled with material in a way that it sustains the given loads best. The loads are applied on a lower edge of the design space while the structure is fixed on the opposing side.
The above goal definition indicates that the global deflection of the structure, which is measured by the global elastic strain energy $C = \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{K}\mathbf{u} = \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{f}$ (C for *compliance*), should be minimal. Minimizing elastic strain energy can be interpreted as maximizing the stiffness of the structure which is a common goal in vehicle design. The following sections treat all necessary steps that have to be considered when setting up a typical topology optimization like this.

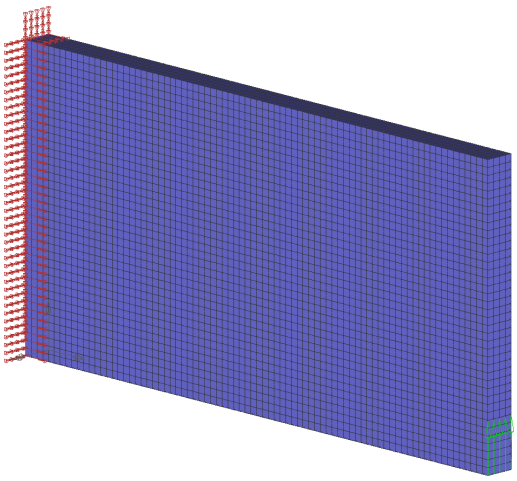


Figure 3.2: *Design space of case study*

3.1.3.1 Boundary conditions and loads

The meshed design space is fixed at one end by prescribing zero displacement for all three (translational) degrees of freedom of the nodes at this boundary. The external load is distributed via nodal point forces on an opposing edge. Obviously, the choice of boundary conditions and loads defines the force flow within the structure and therefore has a major influence on the final result.

3.1.3.2 Goal and constraint functions

The goal for this topology optimization is to maximize stiffness. This is equivalent to minimizing the global elastic strain energy (compliance) $f = C = \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{K}\mathbf{u} = \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{f}$. Therefore, compliance is defined by `$DCONSTRAINT COMPLIANCE` first and then set as the goal function with `$DOBJECT`.
A maximum modification of the design variables, i.e. element filling ratios, in each iteration is restricted to 0.1 by the parameter `$DSVMLIM` to avoid too large steps. Design variables limits are specified by `DSVLIMIT`: a lower design variable limit of 0.01 is defined for numerical reasons and an upper limit of 1.0, of course.
`$DSVFILT` defines a minimum member size of 0.05. The mass/volume constraint prescribing 50 % of the design space is achieved by the parameter `$DSVVAL`.

```
$SYSTEM
   $DSVVAL
    DESET.1   0.5
$END SYSTEM
$MODIFICATION
   $DSVLIMIT
    DESET.1   0.01   1.0
   $DSVMLIM
    DESET.1   0.1
   $DSVFILT TYPE = TOPO
    1   5.0
   $DCONSTRAINT COMPLIANCE  TYPE = GLOBAL  SITUATION = STATIC  LPAT = ALL  DATTYPE = REAL
    COMPL   :   0.0   NO
   $DOBJECT CONSTRAINT = COMPL
$END MODIFICATION
```

3.1.3.3 Local approximation methods

The SIMP method with a default penalty factor $p = 3$ (see Ch. 2.1.7.1) is selected by specifying `KIND = POLY` for the `$DVTPAR` parameter. Also, the filling ratios are allowed to vary during the optimization process.

```
$SYSTEM
   $DVTPAR  DEID = 1  KIND = POLY  FILL = VAR
    3.0
$END SYSTEM
```

Further parameters related to topology optimization can be specified. To begin with, there are the tolerance for convergence of design objective, `TOLOBJ`, and the maximum acceptable constraint violation, `CSTVIOL`. If there are no obvious reason, the default values of 5.0×10^{-3} and 2.0×10^{-2} can be used.
A minimum member size, i.e. checkerboard filter radius, is set by `CFILTER`. Alternatively, a minimum dimension size can be specified by using the parameter `$DSVFILT`.
In order to investigate the influence of the approximation scheme (`METHOD`), all available methods are used and compared regarding their result and computational effort which is summarized in Ch. 4.1.1. A complete list of methods is given in Tab. 3.2. It is noted that all methods are local approximation methods.

Method	Approximation type
GCA	Generalized convex approximation (default)
SCP	Sequential convex programming
SLP	Sequential linear programming
SQP	Sequential quadratic programming
SCPS	Sequential convex programming with sparse constraints
SLPS	Sequential linear programming with sparse constraints
SQPS	Sequential quadratic programming with sparse constraints

Table 3.2: Local approximation methods for topology optimization in PERMAS

3.1.4 Shape optimization

The exemplary problem for shape optimization deals with the modification of a transition radius of a shaft that is subjected to external loads where the goal is to reduce stresses, see Fig. 3.3. The von Mises equivalent stress is used as a stress measure of this three-dimensional load case and an upper limit is defined as constraint. The goal is to fulfil this stress limit at minimal weight.

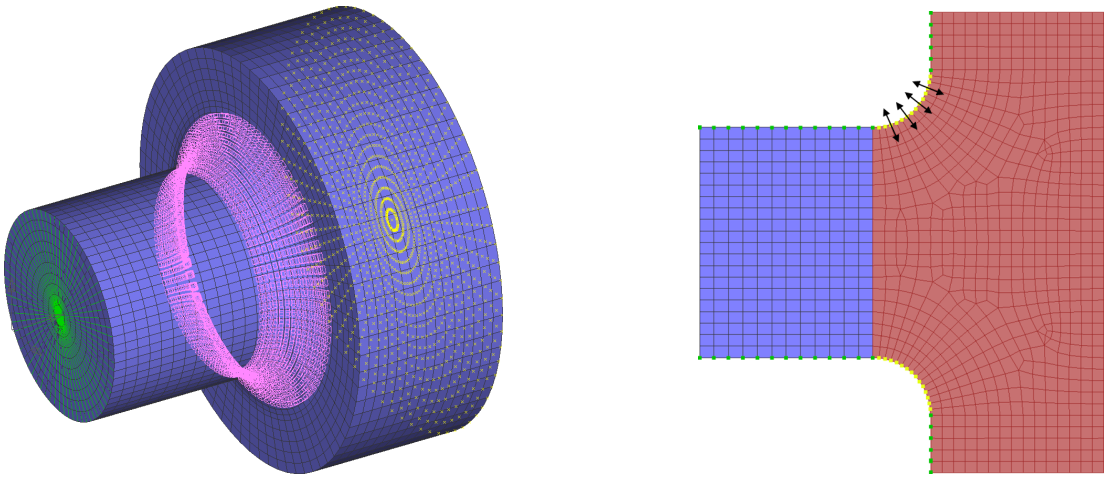


Figure 3.3: *FE model for shape optimization*

3.1.4.1 Design space

In the right part of Fig. 3.3, the design space is highlighted in red. Here, the nodal coordinates are smoothly interpolated according to the shape modification while the rest of the model is fixed. The displacement during the optimization accounts for radial symmetry of the shaft (parameter `$DESYMM`) and a smooth transition (C^2-continuity, `$DERESTRAINT`) to the bearing which is located on the smaller diameter.

3.1.4.2 Boundary conditions and loads

The shaft is clamped at the end with the larger diameter by prescribing zero displacement of all nodal degrees of freedom that lie in this cross-section. The external load consists of both a transversal force and an axial torque and is introduced via a RBE2 element that is connected to the cross-section of the smaller diameter.

3.1.4.3 Goal and constraint functions

The goal function (`$DOBJECT`) is minimal weight subject to fulfilled element stress constraints (`$DCONSTRAINT ELSTRESS`). Hereby, only a set of elements that lie in the region of the transition radius are checked to reduce analysis time.
The design nodes are selected from the transition radius and their displacement is defined to be orthogonal to the boundary (`$DVCOOR`), see Fig. 3.3. The initial displacement (`$DSVVAL`) is set to zero, the limits (`$DSVLIMIT`) and the maximum step width per iteration (`$DSVMLIM`) are assigned reasonable values.

```
$SYSTEM
   $DSVVAL
    1   0.0
    2   0.0
!   ...
   $DVCOOR  KIND = DIRECT
    101   1   SURFACE = 1   :   1.0   :   1
    102   1   SURFACE = 1   :   1.0   :   2
!   ...
$END SYSTEM
$MODIFICATION
```

```
   $DSVLIMIT
    1    -2.0    2.0
    2    -2.0    2.0
!   ...
   $DSVMLIM
    1    0.1
    2    0.1
!   ...
   $DCONSTRAINT ELSTRESS  TYPE = SET  SITUATION = STATIC  LPAT = ALL  DATTYPE = REAL
    STRESS   :   STRESS_SET   2   NO   300   :   100
   $DCONSTRAINT WEIGHT  TYPE = GLOBAL  SITUATION = STATIC  DATTYPE = REAL
    WEIGHT   :   NO   2.0E-04
   $DOBJECT CONSTRAINT = WEIGHT
$END MODIFICATION
```

3.1.4.4 Local approximation methods

All shape optimization performed within this work are FEM-based and therefore use shape basis vectors to modify the structure. These shape basis vectors are calculated by PERMAS automatically after specifying the design space, design nodes, their degrees of freedom and modification limits. In order to check if the obtained basis shape vectors are reasonable, they can be visualized in the PERMAS pre-processor VisPER prior to running the analysis.

A number of parameters for performing a shape optimization (`OPTIM`) can be specified by the user. First, there are the tolerance for convergence of design objective, `TOLOBJ`, and the maximum acceptable constraint violation, `CSTVIOL`. If there are no obvious reason, the default values of 5.0×10^{-3} and 2.0×10^{-2} can be used.

Next, a tolerance for convergence of design variables can be specified if needed. However, this is turned off by default and only needs to be used when there are issues with the first tolerance parameter `TOLOBJ`.

The calculation of the design sensitivity is specified by the parameter `DSMETHOD` with the options `AUTO, DIRECT, ADJOINT` and `NONE`. It is set to automatic selection to let the solver decide which method is beneficial. In the special case of non-linear stress result combination, shape optimization only works with `NONE` which refers to perform the sensitivity analysis based on finite differences.

As in topology optimization, a variety of local approximation schemes (`METHOD`) is available, see Tab. 3.3. Again, they are compared in terms of result and computational cost.

Method	Explanation
AUTO	Automatic selection from methods below (default)
GCA	Generalized convex approximation
SCP	Sequential convex programming
SLP	Sequential linear programming
SQP	Sequential quadratic programming
WLIN	Solve by Wedge-constrained LINear programming scheme
LDR	Solve by Locally improved DiRect (Dividing Rectangle) scheme

Table 3.3: Local approximation methods for shape optimization in PERMAS

3.2 Structural optimization of an engine mount bracket

During the preceding academic example, basic knowledge about how to set up a structural optimization could be gathered. This is now applied to a real case where the goal was to propose a new engine mount bracket design. This is combined with related analysis to evaluate the new component and to carry out a comparison with the current bracket design.

3.2.1 Overview of analysis steps

The starting point are topology optimizations where different variants with reduced mass compared to the current design are obtained. Hereby, the finite element model only consists of the design space and therefore does not consider surrounding components. This is due to the fact that an extended engine model is created later where stress calculations come into focus.
The weight variants are compared in terms of stiffness with the existing bracket in a linear static analysis. Also, a preliminary modal analysis is performed in order to identify a possibly critical dynamic behaviour that could lead to rejection of the new design at an early stage.
After approval, the variant with comparable stiffness but lower mass is transferred into a "smooth" CAD model. Again, a linear static stress analysis is performed but the finite element model now includes all components which are connected to the engine mount bracket, even tightened bolts. Based on the stress results for a given load history, a multi-axial fatigue analysis is performed using the software FEMFAT. The identified critical spots are then addressed in a shape optimization. For the modified design, a final modal and linear buckling analysis is performed.

3.2.2 Topology optimization

The topology optimization covers the creation of the finite element model, i.e. meshing, defining boundary conditions and loads. Furthermore, the analysis procedure that includes an iterative topology optimization based on a linear static analysis has to be specified.

3.2.2.1 Design space and meshing

At first, the design space for the new component has to be defined. This is done in cooperation with the design department that can export a CAD model of the available space in the engine compartment, see Fig. 3.4. In the illustration on the right, three through holes for bolted connections can be identified. The upper two bolts are screwed into the crank case while the lower bolt is screwed into the upper part of the oil pan. The left picture shows the connection to the engine bearing on the opposing side of the design space. When the design space is derived, it is of great importance to already consider accessibility for assembly.

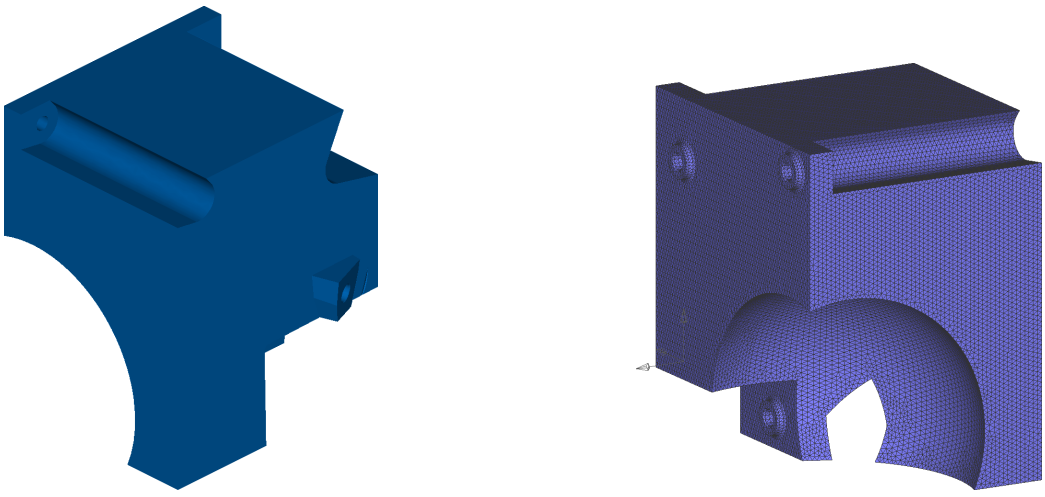


Figure 3.4: *Design space, unmeshed (left) and meshed (right)*

The geometry of the CAD model is imported a FEM preprocessor for meshing. For complex geometries like this, meshing within a reasonable amount of time is only possible with automatic tetrahedral meshing that is offered by most pre-processors. First, a surface mesh is created by using tria elements. Afterwards, a volume mesh with tetrahedral elements is derived.

The average element size is oriented towards the minimal feature size that can be manufactured. In this case, this parameter is defined by the casting restrictions that specify a minimal wall thickness of few millimetres.

The surface mesh is created with the pre-processor *SimLab* (Altair Engineering, Inc.) that enables high-quality meshes and further allows special treatment of geometry features. Here, all fillets have mapped meshes and a maximum angle per element of 30 ° over the fillet is defined for a good geometry approximation. This is especially of significance since stress peaks are usually found on the surface of fillets. Furthermore, a maximum aspect ratio of 10 is prescribed.

The surface mesh is then transferred into a volume mesh with PERMAS linear tetrahedral elements `TET4`[1] by using the pre-processor *MEDINA Pre* (T-Systems). This is because all further settings of the finite element model can be done conveniently with this programme.

3.2.2.2 Boundary conditions and load cases

For topology optimization, surrounding components, i.e. the crank case, the upper part of the oil pan and the engine bearing, are not included in the FE model yet. Therefore, nodal RBE2 elements are used for defining constraints and the introduction of loads, see Fig. 3.5. The left illustration shows one of the three RBE2 elements connected to the contact surface of the bolted connections. The translational displacement of these rigid elements is fixed via single point constraints (SPCs). However, rotation is not prevented because the connected components have an elastic behaviour in reality. Fixing all degrees of freedom would lead to an overly stiff representation of the bolted connection which effects the topology optimization results drastically as was learned from first optimization attempts.

As the right illustration depicts, loads are introduced via a force measurement point from vehicle testing experiments. The point is located inside the indicated solid core of the engine bearing. The dependent nodes of the RBE2 element are those located on the contact surface to the engine bearing.

Obviously, the engine mount underlies loads in all directions during driving. As mentioned earlier, the absolute value of defined loads is not important here. In multi-axial load cases, only the weighting of the single loads counts and therefore units loads of 1 kN are defined in all three spatial directions. In the theory parts, different methods of handling the weighting of loads were described. The last and most advanced method from Eq. 2.42 is chosen. Since this non-standard condition is not implemented in PERMAS, a Fortran subroutine was programmed for that purpose (see App. C.1.3).

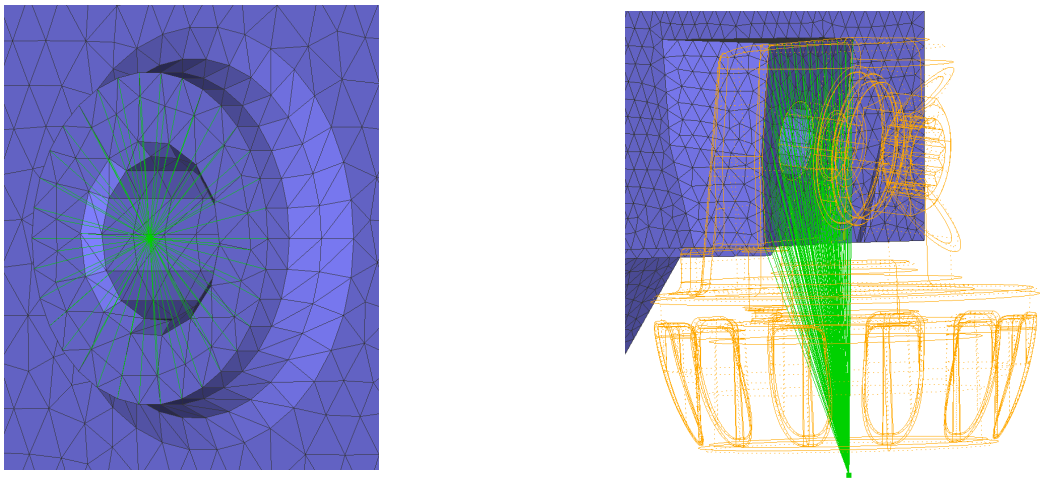


Figure 3.5: *RBE2 elements for boundary conditions (left) and load introduction (right)*

[1] Linear `TET4` elements are selected after the case studies showed that they perform as well as their more costly, quadratic counterparts for topology optimization.

3.2.2.3 Goal and constraint functions

The objective that has been set is to obtain a structure that is at least as stiff as the current one but at reduced weight. Thus, the goal is to minimize compliance, i.e. global elastic strain energy. The necessary weight constraint is set to $1.0 \cdot m_0$, where m_0 is the weight of the current engine mount bracket. Afterwards, the allowable weight is decreased in steps of $0.1 \cdot m_0$ until the structure is less stiff than its competitor. In brief, we have an optimization problem that has been stated in the theory chapter already but is repeated here for convenience:

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) = C(\mathbf{x}) = \frac{1}{2}\mathbf{u}^{\mathrm{T}}(\mathbf{x})\mathbf{f} = \frac{1}{2}\mathbf{u}^{\mathrm{T}}(\mathbf{x})\mathbf{K}(\mathbf{x})\mathbf{u}(\mathbf{x}) && \mathbf{x} \in \mathbb{R}^{n_{\mathrm{el}}} \\ &\text{subject to} && g(\mathbf{x}) = m(\mathbf{x}) = \sum_{j=1}^{n_{\mathrm{el}}} (\rho_j(\mathbf{x}) V_j) \leq k\, m_0 && j = 1, \ldots, n_{\mathrm{el}} \quad , \quad k = 1, 0.9, 0.8, \ldots \end{aligned} \tag{3.1}$$

Furthermore, a minimum member size is defined to account for a minimum wall-thickness by using the parameter `$DSVFILT TYPE = TOPO`. A demoulding direction with shrinking of material cross-section in both directions can also be specified by defining `$RELDIR SHRINK = BOTH`.

3.2.2.4 Evaluation and comparison

The topology optimization results are evaluated and compared to the current engine mount bracket in terms of compliance. Starting from an optimization with a weight constraint of $100\,\%\, m_0$, the weight limit is reduced until the compliance value is just above the reference from the current bracket. This variant then is a candidate for the new design draft.

One important learning step could be achieved regarding the compliance values that are a free by-product in each optimization loop because they define the goal function. The used method for topology optimization here is the so-called SIMP method that assigns a filling ratio x_i to each element, see Ch. 2.1.7.1. The mass of the element is linearly dependent on x_i while the element stiffness matrix is not: $\mathbf{E} = \mathbf{E}\,(x_i^p)$. For a regular linear static analysis, the penalty factor p should take a value of around 3.
Consequently, the stiffness matrices of all elements with intermediate filling ratios $0 < x_i < 1$ is penalized significantly stronger than the mass density. For example, a filling ratio of 0.8 will lead to a density $\rho = 0.8\,\rho_0$ but an element stiffness matrix of only $\mathbf{E} \approx 0.5\,\mathbf{E}_0$. Since for numerical reasons, a distinct 0/1-distribution of filling ratios cannot be achieved, every single element is too soft in topology optimization.
To overcome this method dependent problem, a subsequent linear static analysis has to be performed where only fully filled elements are included in the model and all remaining ones are deleted from the design space. This procedure is described in the next paragraph.

Since the output of the topology optimization are the element filling ratios x_i, the user has to specify a cut-off limit $0 < x_{\mathrm{co}} < 1$ such that all elements with a filling ratio lower than that are not displayed. This on the other hand means that the (intermediate) mass contributions from all those hidden elements is "removed" from the model whereby the displayed elements can be seen as fully filled elements with a filling ratio $x = 1$. From experience, a cut-off limit between 0.4 and 0.5 will display as much elements as needed to reach the specified weight limit for a reasonably converged problem. These selected elements are then exported as a new finite element model and a linear static analysis is performed to calculate the real compliance (global strain energy) values. These will be reduced now because only elements with stiffness matrices that were not penalized are considered in the problem.
As an essential part of evaluation, cutting off "low-density elements" reveals the proposed topologies inside the design space. Their visualization by using the post-processor *MEDINA Post* (T-Systems) can be a helpful mean of reviewing for plausibility.

Furthermore, the convergence curve of the goal function and the constraint violation for each iteration has to be checked. As experience shows, the curve can be quite irregular within the first iterations, especially when there is a constraint violation. As soon as the weight constraint is reliably fulfilled, the goal function should convergence strictly towards its optimum value until the convergence criterion is reached.

3.2.3 Derivation of a CAD model

Once a weight-optimized variant is selected, a "smooth" CAD model has to be deduced. Therefore, the finite element model is reduced to its boundary using tria elements. This model is then exported as a STL file that can be read in by the CAD software CATIA.
Due to the organization structure at Porsche, CAD design is not done by calculation engineers but commissioned at the design department. Hence, the derivation of a CAD model was not done by the author but by a design engineer who sent back the CAD model upon finishing.

3.2.4 Extension from an engine mount bracket to a half engine model

Once, the CAD model of the new engine mount bracket has been received from the design department, a new finite element model is created from scratch that contains additional components, see Tab. 3.4. This step could have been done earlier but the CAD data for the remaining engines parts have not been available from the start. However, since stresses on the contact surfaces as well as eigenfrequencies for modal analyses have to be calculated accurately now, this model extension is considered inevitable.
It is furthermore assumed that the engine is symmetrical such that the crank case and oil pan are cut in half and fixed in the symmetry plane $y = 0$.

3.2.4.1 Meshing

First, all parts are meshed independently by use of the pre-processor *ANSA*. Here, quadratic tetrahedral `TET10` elements are used for accurate stress calculation. The engine mount bracket has a very fine mesh in order to allow a good geometry approximation and reliable stress predictions. The element size is chosen such that there are at least three elements over the smallest wall thickness.
All remaining parts have a coarse mesh to keep the size of the model as small as possible which allows short calculation times. Especially in an iterative process such as optimization, additional analysis cost effects every iteration and therefore should be avoided if possible. In some areas, small geometric details such as chamfers cannot be represented by the coarse mesh. This also leads to a small number of bad shaped elements. Since these are far away from the engine mount bracket itself, no noticeable effect is expected. However, the mesh size in connection regions to the engine mount bracket is adapted to its fine mesh for a good transition.

Component	**Material**	**Number**
Engine mount bracket	Aluminium alloy (castable)	1
Crank case (right half)	Aluminium alloy (castable)	1
Upper part of oil pan (right half)	Aluminium alloy (castable)	1
Lower part of oil pan (right half)	Aluminium alloy (castable)	1
Solid core of engine bearing	Steel	1
Bolt type A (bracket - crank case)	Steel	2
Bolt type B (bracket - upper part of oil pan)	Steel	1
Bolt type C (bracket - engine bearing)	Steel	1

Table 3.4: Components of extended half engine model

3.2.4.2 Contact definition

The existence of different components in the model requires the definition of contacts that prevent penetration of finite elements which does not reflect reality. These contacts cause reaction forces when active. A special case of contact are the bolted connections which define a cylindrical thread coupling with pretension in axial direction and optionally taking into account radial spreading and axial torque caused by the thread's flank/pitch geometry. These special contact definitions and all further settings are defined in the pre-processor *MEDINA Pre* and by manually editing the PERMAS input file, respectively.
In order to obtain good results, all connections to the engine mount brackets have compatible meshes which enables the definition of so-called *PERMAS contacts*. If a contact is active, a corresponding normal force is applied on the model. However, no friction that causes in-plane forces and possibly slipping is considered here.

This would drastically increase the computational cost of the contact analysis. Instead, in-plane multi-point constraints (MPCs) are defined. After the analysis, MPC in-plane forces are compared to PERMAS contact normal forces and, based on a friction coefficient for the material combination, it can be estimated if slip would occur at all. If so, a repeated analysis with enabled friction should be performed. In comparison to the use of incompatible meshes this yields better results and experience shows that compatible meshes have less numerical problems.

All other connections of the model, e.g. between the crank case and the oil pan, do not use compatible meshes because a detailed contact analysis is not necessary here and consequently, time and effort can be saved. Instead, *MPC isurfaces* are defined that couples selected nodes from one component to a defined surface of the opposing component (node-surface coupling).

Given these contact definitions, the model is assembled completely. Nevertheless, one-dimensional spring elements with a very small spring coefficient of $1\,\mathrm{N}\cdot\mathrm{mm}^{-1}$ are created to connect all individual components. This is necessary because PERMAS does not initiate all contact definitions from the start and in this state, the finite element model can disintegrate which disposes the solver to exit on error.

3.2.4.3 Boundary condition and load history

The increased complexity of the model also effects boundary conditions and loads. All nodes which are located on the symmetry plane $y = 0$, i.e. crank case and oil pan nodes, are fixed. Owing to contacts and preloaded bolts, the simple load cases from topology optimization are replaced by a load history with sequenced load cases. The following table summarises a complete load history where all known load cases are integrated.

Load pattern	$t = 0$	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$	$t = 6$	$t = 7$
#0: PERMAS contacts	1	1	1	1	1	1	1	1
#1: Pretension threads (bolts 1-3)	0	1	1	1	1	1	1	1
#2: Pretension threads (bolt 4)	0	0	1	1	1	1	1	1
#3: Load in $-x$ ("soft" crash)	0	0	0	1	0	0	0	0
#4: Load in $+y$ (right corner)	0	0	0	0	1	0	0	0
#5: Load in $-y$ (left corner)	0	0	0	0	0	1	0	0
#6: Load in $+z$ (bump)	0	0	0	0	0	0	1	0
#7: Load in $-z$ (bump)	0	0	0	0	0	0	0	1

Table 3.5: General load history ("0" = inactive, "1" = active)

3.2.4.4 Output variables

The requested output for a linear static analysis are shown below and are given for every single time increment, see Tab. 3.5. As a consequence of the increased number of degrees of freedom, solving these jobs locally on workstations requires too much solving time or the computation might fail directly due to too little scratch disc space. So, all following jobs are only computed on Porsche's supercomputing resources using 16 Intel Xeon CPUs with 12 GB of memory each by default.

- Displacement vector u_i
- Element stress tensors $\sigma_{\mathrm{el},ij}$
- (Interpolated) nodal point stress tensors $\sigma_{\mathrm{nod},ij}$
- Contact analysis quantities (contact status, contact pressure, etc.)
- MPC quantities (normal forces, in-plane forces, etc.)
- Reaction forces R_i

3.2.5 Stress and multi-axial fatigue analysis

The multi-axial fatigue analysis follows up the linear static stress analysis and is performed with the software *FEMFAT*. Since we are only interested in the fatigue behaviour of the engine mount bracket, the operational strength analysis is restricted to this component while all other parts are not considered.

The linear static analysis takes only loads into account which are relevant for operational strength. This means that the load history only includes both cornering loads #4 and #5 besides the initial assembly steps #0, #1 and #2. Tab. 3.6 shows the reduced load history.

Load pattern	$t = 0$	$t = 1$	$t = 2$	$t = 3$	$t = 4$
#0: PERMAS contacts	1	1	1	1	1
#1: Pretension threads (bolts 1-3)	0	1	1	1	1
#2: Pretension threads (bolt 4)	0	0	1	1	1
#3: Load in $+y$ (right corner)	0	0	0	1	0
#4: Load in $-y$ (left corner)	0	0	0	0	1

Table 3.6: Load history for multi-axial fatigue analysis ("0" = inactive, "1" = active)

FEMFAT requires model data including all element definitions and their corresponding nodal coordinates which are exported from the half engine model. Next, transient element stress results from all relevant time increments have to be exported for FEMFAT, too. Thus, data that is imported into FEMFAT is summarized in Tab. 3.7.

Data type	**Quantities**
Node list	nid
Element list	eid, list(nid)
Nodal coordinates	x_{idof}
Element stress tensor	$\sigma_{ij}(t = 2, 4, 5)$

Table 3.7: Finite element data imported into FEMFAT

After all finite element data are read in, material properties have to be specified. In this case, the material of the engine mount has already been tested by an external laboratory and the results can be imported by use of a material file. The properties needed here are summarized in Tab. 3.8.

The requested output of the analysis is the safety factor SF_A, of course. For further post-processing, the calculated stress amplitude σ_a and mean stress σ_m, the local fatigue limit and the influence factor of the stress gradient on fatigue strength are also written out.

3.2.6 Shape optimization

The goal of this section is to identify critical spots from the operational strength safety analysis and address these in a shape optimization where the half engine model is used again.

3.2.6.1 Identification of critical spots

The critical spots (red) with safety factors $SF_A < SF_{A,\min}$, see Ch. 2.2.4.1, can easily be visualized by importing the FEMFAT output file into the post-processor *MEDINA Post*, see left part of Fig. 4.10. Red areas are all along the lower edge of the contact surface to the engine bearing. This, however, is a result of the locally over-predicted element stresses which are usually found on edges of contact surfaces. Since these stresses are not physical, they are not considered here.

Another critical spot is on a fillet to the right of the contact surface and another one on a fillet further below. These two spots are of real concern and have to be modified by means of a shape optimization.

General data	Elastic Poisson's ratio ν, material class (aluminium casting alloy)
Linear static data	Young's modulus x-direction E_x
Cyclic stabilized data	Cyclic hardening coefficient K', cyclic hardening exponent n', fatigue strength coefficient σ'_f, fatigue strength exponent b, fatigue ductility coefficient ϵ'_f, fatigue ductility exponent c
Strength data, tension	Ultimate strength, yield strength, pulsation endurance strength ($R = 0$), alternating endurance strength ($R = -1$)
Strength data, compression	Ultimate strength, yield strength, pulsation endurance strength ($R = 0$), alternating endurance strength ($R = -1$)
Strength data, bending	Ultimate strength, yield strength, pulsation endurance strength ($R = 0$), alternating endurance strength ($R = -1$)
Strength data, shear	Ultimate strength, yield strength, pulsation endurance strength ($R = 0$), alternating endurance strength ($R = -1$)
S-N data	Type of S-N curve (linear model), slope of S-N curve b, cyclic limit of endurance $N_f = 10^7$, stress limit of endurance S_f, survival probability (... %)

Table 3.8: Material properties for FEMFAT

3.2.6.2 Design space and design variables

The design space for shape optimization is spaciously laid out and exceeds the actual fillet that is indicated in Fig. 3.6. On the upper end, the design space is limited because the surface beyond must remain flat as it serves as a contact surface. No restrictions are given on the other end so that it is extended far down to enable a smooth transition for the fillet modification.

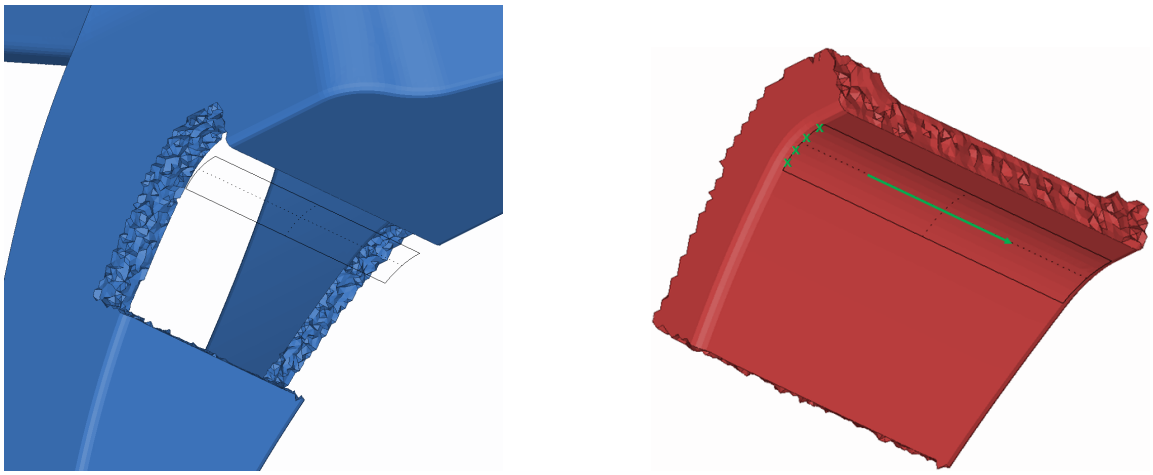


Figure 3.6: *Design element region (red) for shape optimization*

As indicated in the right part of the picture, four nodes from the left boundary of the fillet are chosen to be the design nodes. The displacement of these nodes is defined to be perpendicular to the surface. Together with a definition of a fillet direction, the radius is modified equally over this preferred direction.

Another novelty in this analysis is that for each design node, the design space is modified separately. At the end of an iteration, all four nodal displacements fields are superposed. Compared to the alternative of having only one design space, this yield smoother shape modification at insignificant increased cost (for more detail, see Ch. 2.1.7.2).

3.2.6.3 Goal and constraint functions

As argued in Ch. 2.2.3.2, minimized weight is defined as the goal function. The stress constraint aims to reduce the amplitude stress[2] that has been found too high in the multi-axial fatigue analysis. As discussed earlier, the amplitude stress is estimated by subtracting the "assembly stress state" (time increment $t = 2$) from the combined "assembly+load stress state", see Tab. 3.6. However, there are two combined load cases: time increment $t = 3$ with a negative load in y-direction and $t = 4$ with a positive load in y-direction. The previous multi-axial fatigue analysis showed that a positive load leads to tension in the critical region which results in a lower safety factor than the opposite case. Thus, the stress constraint is defined as $\sigma_a \approx \sigma_{\mathrm{vM}}(t=4) - \sigma_{\mathrm{vM}}(t=2) \leq \sigma_{a,\max}$. The complete optimization problem is stated as follows:

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) = m(\mathbf{x}) = \sum_{i=1}^{n_{\mathrm{el}}} (\rho_j V_j(\mathbf{x})) && , && \mathbf{x} \in \mathbb{R}^{n_{\mathrm{el}}} \\ &\text{subject to} && g_j(\mathbf{x}) = \sigma_{a,j}(\mathbf{x}) \approx \sigma_{\mathrm{vM},j}(t=4) - \sigma_{\mathrm{vM},j}(t=2) \leq \sigma_{a,\max} && , && \mathrm{j} = 1, \ldots, n_{\mathrm{el}} \end{aligned} \tag{3.2}$$

This so-called *result combination* is available since the PERMAS development version 14.10.002 or the newly released maintenance version 15.00. It is defined by the parameter `$COMBINE RESULTS` as shown in the following code snippet of a PERMAS input file.

```
...
   $LOADING  NAME = LOADVAR_1
      $NLLOAD TABLE
!      TIMESTEP   0   1
!      ---------------------------------------
       LPAT = 1   1.  1.        ! ASSEMBLY
       LPAT = 2   0.  1.        ! EXTERNAL LOAD
   $END LOADING
...
   $RESULTS
       $COMBINE RESULTS  RULE = SIGMA_A  TIME = 2.
          SITUATION = STATIC  TIME = 1. FACTOR = 1.      ! +1*LPAT2
          SITUATION = STATIC  TIME = 0. FACTOR = -1.     ! -1*LPAT1
   $END RESULTS
...
   $MODIFICATION
      $DCONSTRAINT NPSTRESS  TYPE = SET  SITUATION = STATIC  TIME = 2.
            &      DATTYPE = REAL
       STRESS  :   DCONSTR_FILLET     2      NO        150.
   $END MODIFICATION
...
```

The data lines under `$COMBINE RESULTS` specify which time increments are added where a `FACTOR` can be applied. Here, the results from time step 1 (assembly+load) multiplied by 1 are added to the results from time step 0 (assembly) multiplied by -1 and stored as time step 2. The defined combination is given the `RULE` name `SIGMA_A` that is called in the PERMAS job file, see below.

```
...
SET NOLINCOMBINE = ON
...
TASK LOOPS = 30
   EXEC
      ACTivate SITuation = STATIC
         STATIC
         GENerate ELement STRESS
         COMBINE ELement STRESS RULE = SIGMA_A
         GENerate Nodal Point STRESS
      ACTivate SITuation = OPT
         OPTIM TOLOBJ = 5.E-3 DSMETHOD = NONE METHOD = SCP
TASK END
...
```

[2]In the this problem, the amplitude stress has a dominating influence on the endurance safety factor as the mean stress is negligible. Thus, the mean stress is not considered for the constraint definition.

The result combination is requested for element stresses (`COMBINE ELement STRESS`). A subsequent generation of nodal point stresses already includes the combined time step 2. In order to enable such a non-linear result combination in the first place, the parameter `NOLINCOMBINE` has to be turned in the preamble of the code.

The value for the stress constraint (here 150 MPA, see first code snippet at the bottom) is based on the maximum amplitude stress $\sigma_{a,\max}$ as computed by FEMFAT for the modified Haigh diagram of the critical node. This limit is further multiplied by an internal safety factor SF that accounts for material scatter and alike.

$$\sigma_{a,\max} = SF \cdot \sigma_{a,\max,\text{FEMFAT}} \tag{3.3}$$

3.2.6.4 Validation of operational strength

Once the optimization is finished and a quick visual check of the shape modification seems reasonable, a repeated linear static stress analysis can be performed. As before, the transient stress states are exported and are read in into FEMFAT to control for a satisfied safety condition.
If $SF_A < SF_{A,\min}$, then the stress constraint has to be reduced further and the optimization has to be started all over again.

3.2.7 Concluding analyses

Given the case that all critical spots of the structure could have been eliminated, a modal and linear buckling analysis conclude the complete evaluation of the component.

3.2.7.1 Modal analysis

In order to compare the new engine mount brackets design with the current component, a half engine model is created for both components. The modal analysis is then performed for both brackets with identical boundary conditions. The first six eigenmodes are requested whereby the first one most important, naturally. A code snippet of the PERMAS job file is shown below.

```
...
TASK
   EXEC
      VIBration ANalysis  MODES = 6  FREQLIM = 1000
...
TASK END
...
```

Most importantly it has to be assured that $\omega_1 > \omega_{\text{crit}}$ for the new engine mount bracket, where ω_1 is the lowest eigenfrequencies and ω_{crit} an estimated critical value that is provided by the NVH department of Porsche.

3.2.7.2 Linear buckling analysis

Finally, linear buckling is investigated and again, the first six eigenmodes are requested. This analysis has to prove that buckling does not occur for magnitudes of loads that can occur during operation.
Therefore, a linear static analysis is performed first and the normalized loads are scaled afterwards until buckling of the structure occurs, see below.

```
...
TASK
   EXEC
      ACTivate SITuation = STATIC
         STATIC
      ACTivate SITuation = BUCKLING
         BUCKling ANalysis  MODES = 6
...
TASK END
...
```

4 Results and Discussion

The following chapters elaborate on analysis results and their discussion. First, Ch. 4.1 treats the outcome of the two study cases for topology and shape optimization that are used to investigate how to perform numerical structural optimization in PERMAS. Next, the results of a complete structural optimization of an engine mount bracket are discussed in Ch. 4.2. This includes aspects of finite element modelling, topology and shape optimization as well as multi-axial fatigue, modal and linear buckling analysis.

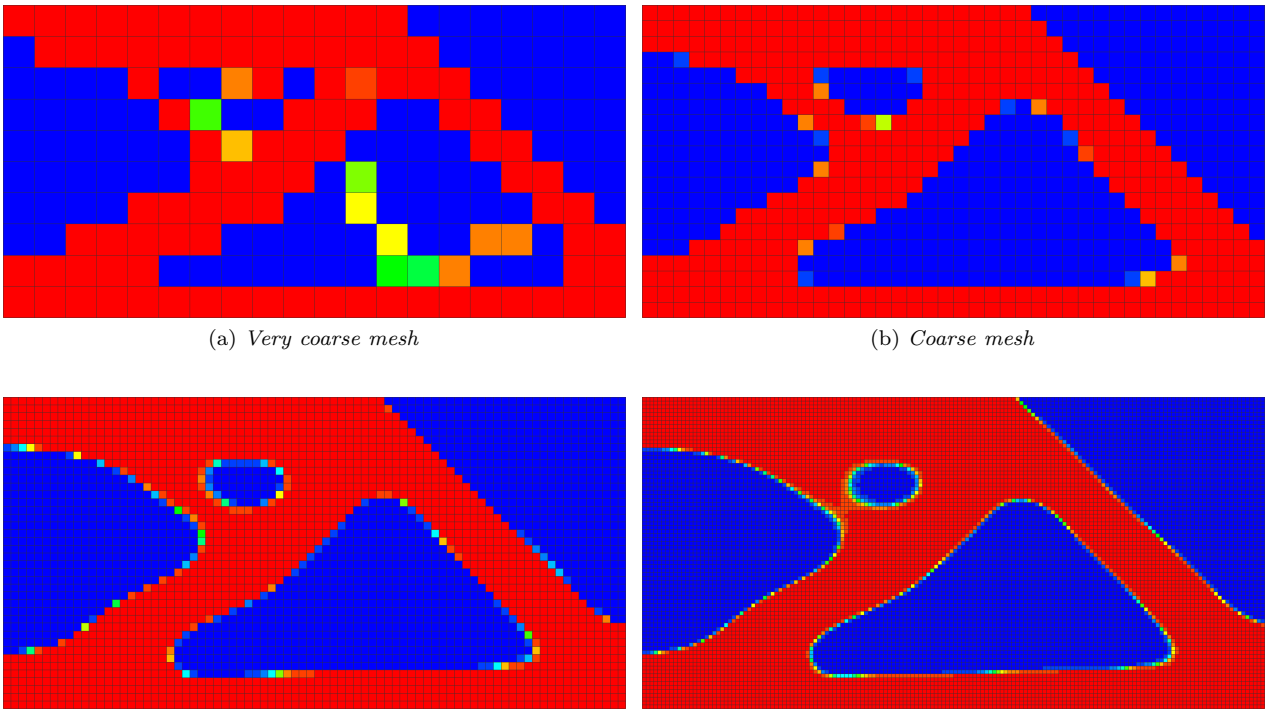
4.1 Settings for structural optimization with FE software

Two academic examples are used to examine different settings and model definitions, one for topology optimization and one for shape optimization. This includes the question which finite elements and mesh resolution should be chosen in the first place. For topology optimization, the influence of the material parameter E for a linear-static analysis is assessed. Furthermore, all additional settings regarding optimization that are available in PERMAS are studied.

4.1.1 Topology optimization

4.1.1.1 Mesh size and element types

The influence of mesh size is demonstrated by solving an identical problem with different mesh sizes. For all analysed element types, i.e. linear `TET4, HEXE8` and (reduced) quadratic `TET10, HEXE20`, the conclusion is the same. Even for a coarse mesh, the basic distribution of material is predicted the same way as for a very fine mesh, see Fig. 4.1 for `HEXE8` elements. A complete overview of all results can be found in App. B.1.1
This leads to the conclusion that the mesh size should only be as fine as needed in order to reduce computational cost that increases rapidly as the mesh is refined (for more detail, see Tab. 4.3). Thus, finite elements should be small enough to represent the basic geometry fairly well and they should be able to model the smallest feature like a minimum wall thickness of the sought structure.



(a) *Very coarse mesh*

(b) *Coarse mesh*

(c) *Fine mesh*

(d) *Very fine mesh*

Figure 4.1: *Material distribution from topology optimization using different* `HEXE8` *mesh sizes and local GCA approximation*

Interestingly, both linear elements TET4 and HEXE8 perform equally well as their quadratic counterparts TET10 and HEXE20 for this type of topology optimization. Here, minimum global elastic strain energy (compliance) is chosen as goal function together with a weight constraint. If stresses appear as goal or constraint function, then linear TET4 elements should be avoided since they are overly stiff. However, stress evaluation in a topology optimization is questionable in the first place because the resulting structures from topology optimization are not smooth, see Fig. 4.1.

Mesh	Number of nodes n_{nod}	Number of elements n_{el}	Goal function f	Analysis time t
TET4, very coarse	462	1,200	6.8388×10^{-3}	00:00:59.750
TET4, coarse	2,583	9,600	6.8202×10^{-3}	00:02:02.767
TET4, fine	16,605	76,800	7.0746×10^{-3}	00:23:28.605
TET4, very fine	117,369	614,400	7.2231×10^{-3}	02:08:33.911
TET10, very coarse	2,583	1,200	8.6631×10^{-3}	00:01:07.449
TET10, coarse	16,605	9,600	8.0188×10^{-3}	00:02:47.625
TET10, fine	117,369	76,800	7.8623×10^{-3}	00:27:55.779
TET10, very fine	878,577	614,400	7.8272×10^{-3}	04:17:22.124
HEXE8, very coarse	462	200	1.7344×10^{-3}	00:00:56.753
HEXE8, coarse	2,583	1,600	7.1781×10^{-3}	00:01:37.613
HEXE8, fine	16,605	12,800	7.2691×10^{-3}	00:09:05.706
HEXE8, very fine	117,369	102,400	7.3812×10^{-3}	00:31:36.192
HEXE20, very coarse	1,553	200	2.1501×10^{-3}	00:00:56.863
HEXE20, coarse	9,285	1,600	8.2077×10^{-3}	00:01:35.295
HEXE20, fine	62,489	12,800	7.2691×10^{-3}	00:08:22.724
HEXE20, very fine	454,257	102,400	7.9148×10^{-3}	01:13:32.361

Table 4.1: Goal function values and analysis time for different fine elements and mesh resolutions

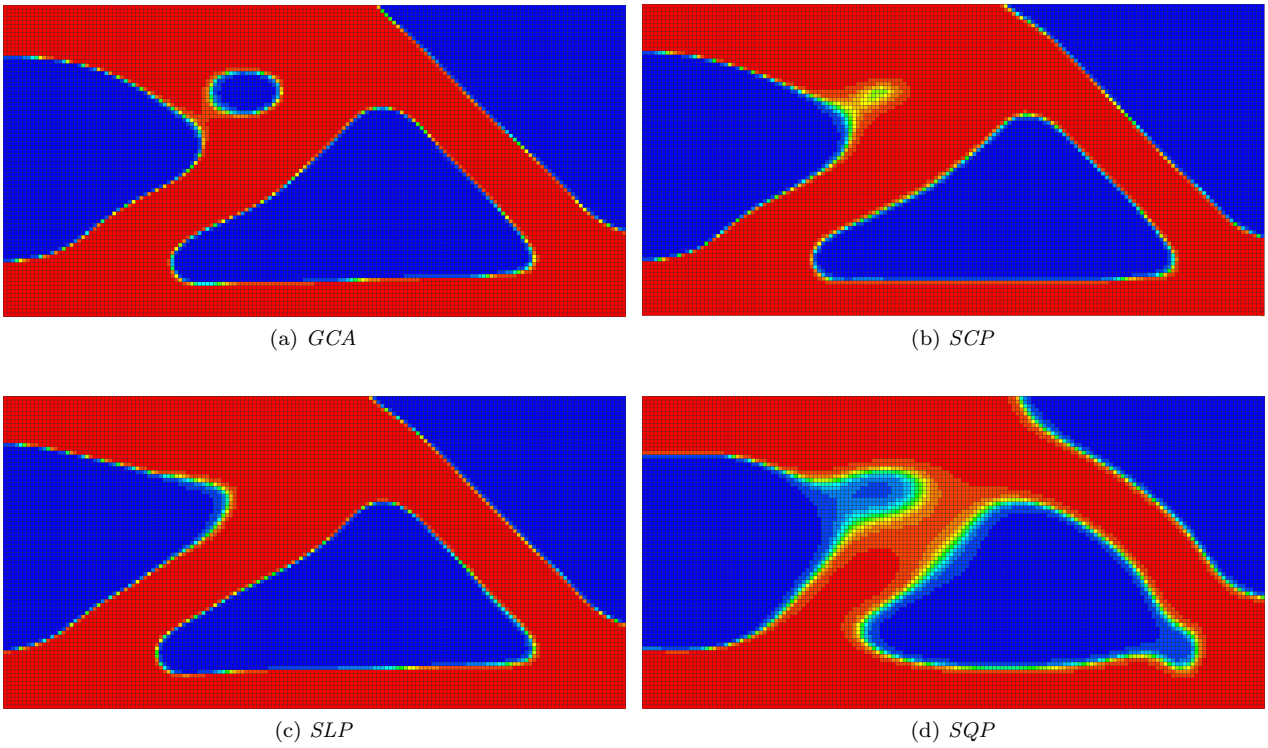
4.1.1.2 Material properties

The dependence on material properties is investigated by a variation of the Young's modulus E for an isotropic material in a linear-elastic analysis. When the goal function is chosen to be compliance as usual, it is found that the topology optimization result is the same for any choice of E. This is because a modification of E does nothing more than linearly scaling the goal function $f(\mathbf{x})$ which changes its value but not the solution $\mathbf{x}$ that minimizes it. Therefore, the topology result is independent on the choice of material.

4.1.1.3 Local approximation methods

All local approximation methods offered by PERMAS, see Tab. 3.2, are compared in terms of result and analysis time. Hereby, a very fine mesh with linear HEXE8 elements was chosen to avoid mesh dependency. Only the standard methods are depicted below. A complete overview that also includes methods for sparse constraints is given in App. B.1.2.
The default method GCA suggest the known framework-like structure with a void in the upper half. SCP and especially SQP suggest a tendency to form a void, too. SLP as the only method does not show any voids at all. For a proper comparison the goal function values that are a measure of "inverse stiffness" are outlined. In addition, analysis times are shown for evaluation computational cost, see Tab. 4.2.

It can be seen that all methods except SQP and SQPS perform similarly well in terms of minimizing the goal function. Leaving out SQP and SQPS for the following comparison, all method require between 20 and 24 iterations for convergence, only GCA needs 30 loops. The slowest methods are the convex methods GCA and SCP while the remaining ones have an approximately 20 % faster convergence.
Given the fact that no clear conclusion can be drawn here, a comprehensive exchange of experience with the PERMAS developers was held. According to their knowledge, the convex methods GCA and SCP are recommended for general purpose and especially when large finite element models with non-linearities (e.g. contacts, minimum wall thickness conditions) are involved.



(a) *GCA* (b) *SCP*

(c) *SLP* (d) *SQP*

Figure 4.2: *Material distribution from topology optimization using different solution methods with a very fine* `HEXES` *mesh*

The special methods SCPS, SLPS and SQPS are intended to be used when there is a large number design variables (typical for topology optimization) together with a large number of "sparse" constraints. Currently, sparse constraints are generated in PERMAS only when both a minimum and a maximum member size[1] is defined [24].

The later topology optimization models for the engine mount include the non-linear definition of a minimum wall thickness and a demoulding direction. However, both a minimum and a maximum member size control is not required. Thus, it is decided to use the methods GCA and SCP for the engine mount bracket and compare them again in terms of their goal function value and analysis time.

Method	**Iterations**	**Goal function (compliance [mJ])**	**Analysis time**
GCA	30	7.3812×10^{-3}	00:31:36.192
SCP	24	7.3982×10^{-3}	00:31:06.239
SLP	23	7.3543×10^{-3}	00:24:26.180
SQP	23	9.3054×10^{-3}	00:24:28.383
SCPS	21	7.4134×10^{-3}	00:22:41.819
SLPS	20	7.3300×10^{-3}	00:21:40.061
SQPS	31	9.7925×10^{-3}	00:32:50.542

Table 4.2: Goal function values and analysis time for different approximation methods

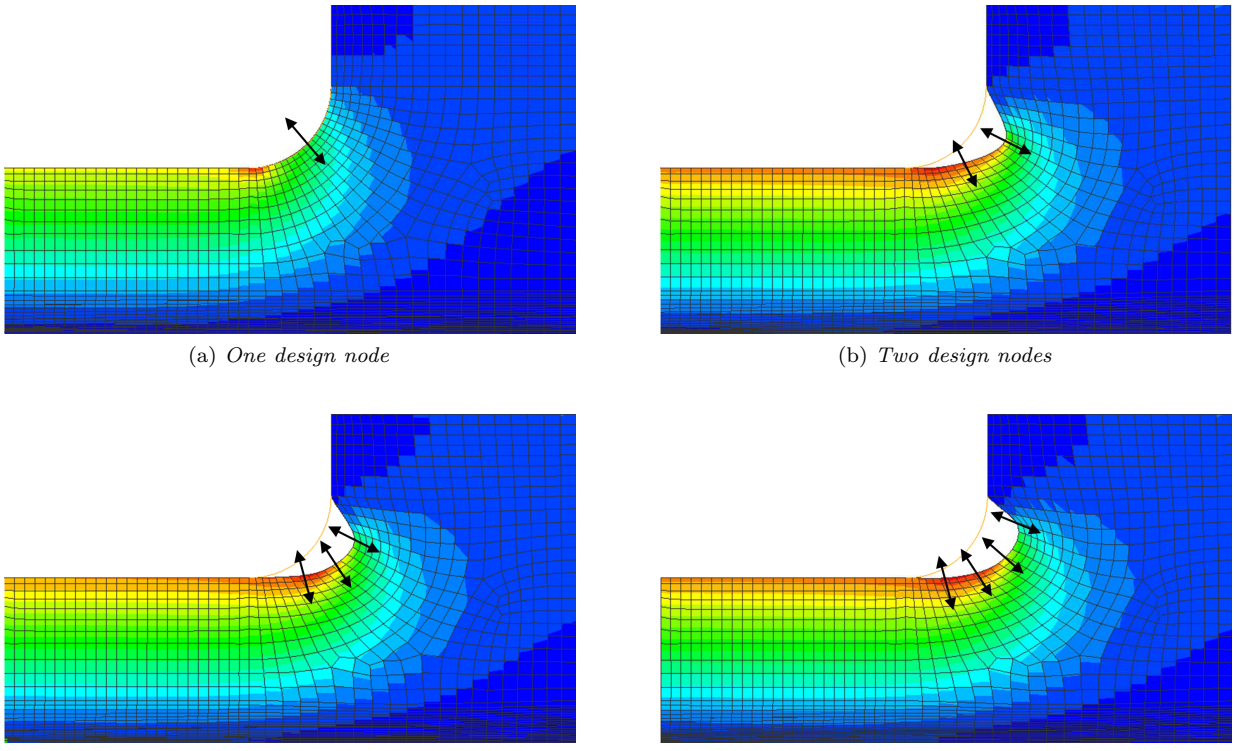
[1] When a topology optimization is performed, it can be favoured that the coherent mass of the resulting structure is not smaller or larger than a certain size. Member size controls defines such limits of the size of connected finite elements.

4.1.2 Shape optimization

The following subsections deal with the results on the number of design variables, mesh size, element types and approximation methods that were investigated for the simple academic example of a transition fillet modification. The outcome is applied on the later shape modification of the engine mount bracket.

4.1.2.1 Number of design variables

First, shape modifications are performed for different numbers of design variables in order to find out how many design nodes are appropriate for a transition fillet modification. A very fine `HEXE8` mesh and the standard approximation method GCA to avoid dependency on the mesh and approximation.



(a) *One design node*

(b) *Two design nodes*

(c) *Three design nodes*

(d) *Four design nodes*

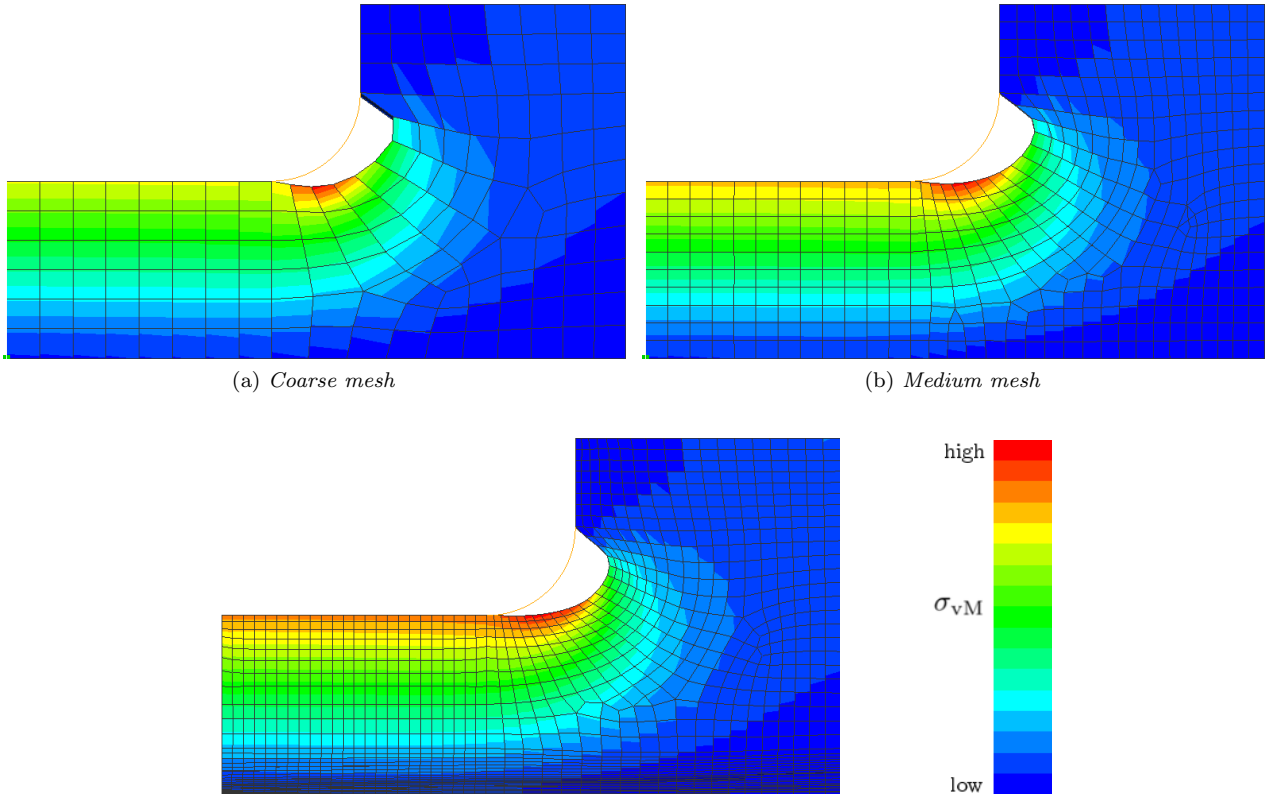
Figure 4.3: *Shape modification and nodal point stresses from different numbers of design nodes with `HEXE8` elements and the GCA approximation method*

The shape modification results are illustrated in Fig. 4.3. Surprisingly, the selection of only one design node does not lead to a solution at all. All other variants deliver suitable results but it is recognized that a growing number of design variables allows for more complex shape modifications which is desirable. However, the solution time increases rapidly as more design sensitivity analyses have to be performed, see Fig. B.8.
In accordance with the PERMAS user support that contributed experience from various problems, a total of three to four design nodes appears as a good compromise between freedom of shape modification and analysis time. For later analyses, it is decided to use four design nodes.

4.1.2.2 Mesh size and element types

The same investigations concerning element size and types is done for shape optimization. Here, three different mesh solutions for linear `TET4, HEXE8` and (reduced) quadratic `TET10, HEXE20` elements are compared.
Again, it can be seen that the basic shape modification is already predicted correctly by a coarse mesh. However, a coarse mesh with only few elements in the modification region does not allow a smooth surface and also the accuracy of stresses that serve as a constraint can be critical.

Therefore, the requirements on the finite element mesh mainly arise from the ability of accurate stress prediction.



(a) *Coarse mesh* (b) *Medium mesh*

(c) *Fine mesh*

Figure 4.4: *Shape modification and nodal point stresses from shape optimization using different mesh resolutions methods with* `HEXE8` *elements and automatic choice of approximation method*

A good mesh for a linear static stress analysis is thus also a good mesh for shape optimization. For this purpose, the use of linear `HEXE8` elements is recommended where possible, otherwise quadratic `TET10` elements.

Linear `TET4` elements should be avoided because of their bad stress prediction. The same applies to quadratic `HEXE20` elements which only perform better than their linear counterparts for coarse meshes but are otherwise very costly and known for their poor contact performance.

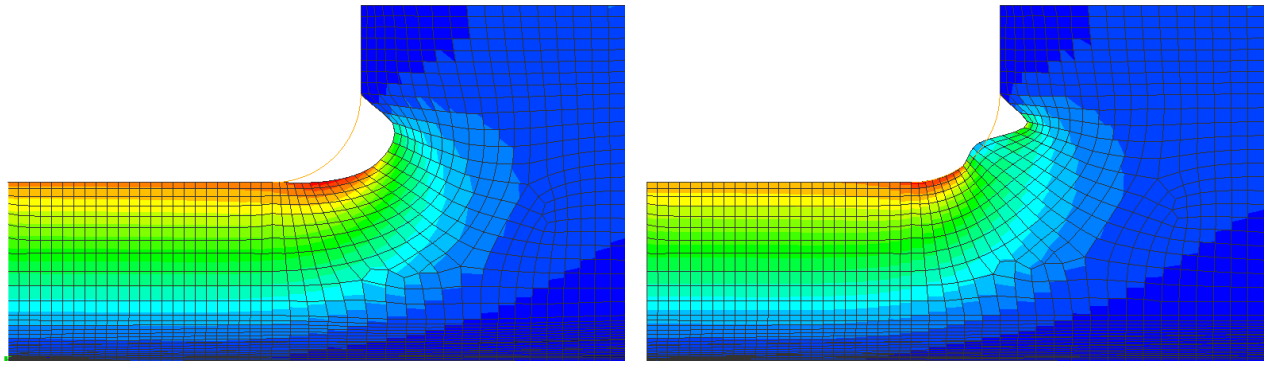
Mesh	n_{nod}	n_{el}	DV1	DV2	DV3	DV4	Analysis time t
`TET4`, coarse	4,656	23,389	0.089	-0.717	-2.000	-2.000	00:04:34
`TET4`, medium	22,430	119,511	0.185	-0.430	-1.616	-2.000	00:55:05
`TET4`, fine	52,835	286,591	-0.111	-0.644	-1.955	-2.000	04:42:07
`TET10`, coarse	34,247	23,389	-0.088	-0.898	-2.000	-2.000	00:05:42
`TET10`, medium	169,744	119,511	-0.151	-0.712	-2.000	-2.000	01:11:36
`TET10`, fine	403,243	286,591	-0.165	-0.757	-2.000	-2.000	05:03:37
`HEXE8`, coarse	6,104	5,616	-0.290	-1.237	-2.000	-2.000	00:01:34
`HEXE8`, medium	35,818	34,272	-0.203	-0.880	-2.000	-2.000	00:11:27
`HEXE8`, fine	109,336	106,084	-0.151	-1.012	-2.000	-2.000	01:00:42
`HEXE20`, coarse	23,906	5,616	-0.269	-1.231	-2.000	-2.000	00:02:05
`HEXE20`, medium	141,690	34,272	-0.150	-0.699	-2.000	-2.000	00:17:05
`HEXE20`, fine	433,018	106,084	-0.132	-0.744	-2.000	-2.000	01:49:29

Table 4.3: Goal function values and analysis time for different fine elements and mesh resolutions

4.1.2.3 Local approximation methods

All available approximation methods for shape optimization in PERMAS, see Tab 3.3, are compared in terms of shape modification which is accomplished by looking at the values of the design variables which correspond to the surface-normal displacement. In addition, the goal function (structural weight) and analysis time is compared for all methods.

The default setting in PERMAS is AUTO. If this is selected, the solver automatically decides which of the available approximation schemes (GCA, SCP, SLP, SQP, WLIN, LDR) should be used. The identical number of iterations and goal function value for AUTO reveals that the solver decided to use the generalized convex method GCA.



(a) *"Identical" results for GCA, SCP, SLP, SQP, WLIN* (b) *Result for LDR*

Figure 4.5: *Shape modification results for different approximation methods*

By comparing all methods visually, it can be seen that all methods except LDR work perfectly fine and deliver identical results to the eye, see above and Fig. B.11. In fact, the differences are so small that the mass difference of the heaviest variant, SQP, to the lighest one, SLP, is only 0.06 %.

A much bigger difference can be seen in terms of calculation time. The slowest method is GCA which also is selected if method is set to AUTO. Therefore, the automatic choice of the solution method means a significant increase of calculation time without benefit.
Almost 50 % less calculation time is needed for WLIN and even a little less for the methods SCP, SLP and SQP. Given this remarkable time advantage, the S*P methods are clearly favourable. The fact that the convex method SCP is also most capable of treating non-linearities such as contact definitions makes it the method of choice for the shape optimization of the engine mount bracket.

Method	**Iterations**	**Goal function (weight [g])**	**Analysis time**
AUTO (=GCA)	7	193.945	01:00:42
GCA	7	193.945	00:58:46
SCP	13	193.944	00:27:38
SLP	14	193.944	00:29:03
SQP	11	194.063	00:24:30
WLIN	28	193.990	00:31:39
LDR	no convergence	no convergence	no convergence

Table 4.4: Goal function values and analysis time for different approximation methods

4.2 Structural optimization of an engine mount bracket

The following chapters cover the structural optimization results of an engine mount bracket whereby previously gained knowledge from both case studies is applied. First, the outcome of the topology optimization and derivation of a smooth CAD model is outlined. This is followed by the extension to a half engine model and a subsequent shape optimization that also includes a multi-axial fatigue analysis. Concluding results are given for modal and linear buckling analyses.

4.2.1 Topology optimization

The first topology optimization is performed with a mass constraint of 100 % m_0, i.e. equal weight as the existing engine mount bracket. Furthermore, a minimum member size that considers a minimum wall thickness for the casting process is taken into account. However, a two-sided demoulding direction that is necessary for the casting process is not specified from the start to assure that all other settings work as they should. Based on this, the material distribution for an optimal stiff design using SCP approximation is shown in Fig. 4.6.

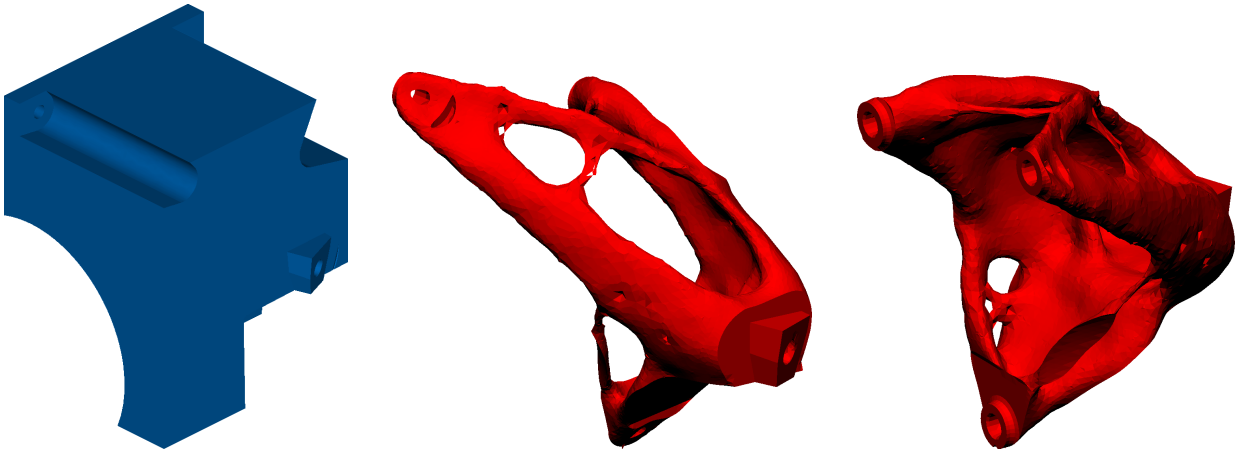


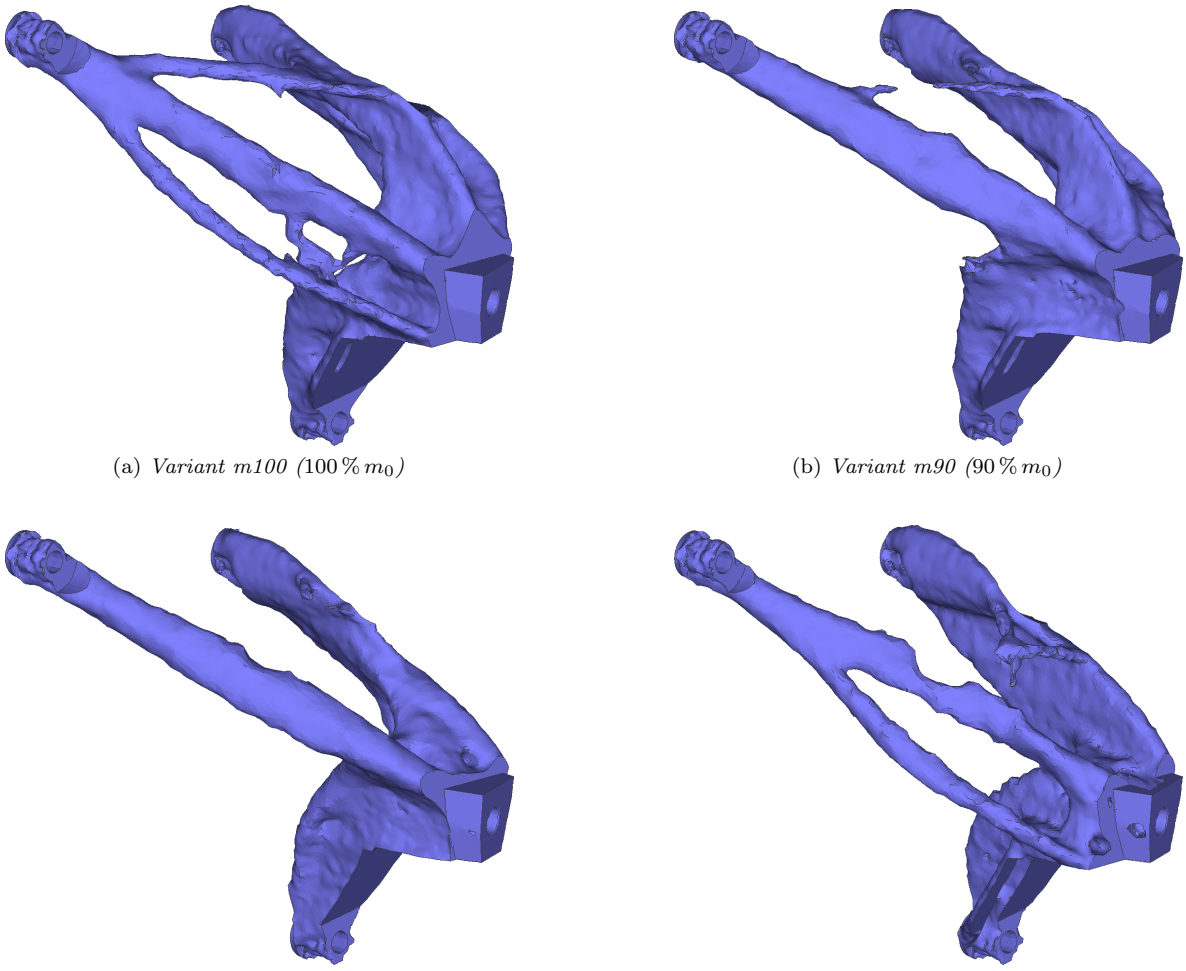
Figure 4.6: *Optimal stiff material distribution inside the design space (100 % m_0, no demoulding direction)*

Most of the material is placed on the direct connection between the three through holes on the engine side and the opposing connection to the engine bearing. This is reasonable because the force flux for loading in x, y or z takes this path inside the design space. On the upper side, a cross-connection can be seen that supports the upper left connection. This is due to its eccentrically location from all other screw connections.

Due to some minor undercuts, this shape is not suitable for two-sided demoulding. Applying a setting in PERMAS that takes this constraint into account however causes numerical problems within PERMAS v14 that surprisingly lead to a completely re-arranged, extrusion-like material distribution with an unsatisfactory convergence of filling ratios, see Fig. C.1. The goal function value is approximately doubled which means a drastic reduction of stiffness. Unfortunately, the source of this problem could not be identified until the end of this thesis and is currently approached by the developers of PERMAS.
As a quick remedy, the same optimization problem is computed with the optimization solver OptiStruct. In this analysis, OptiStruct also uses the SIMP method for topology optimization and the MMA approach for the local approximations for sensitivity analysis. Unlike in PERMAS, dynamic weight factors with modifying values (see Fortran code, App. C.1.3) for the different load cases are not applied. Therefore, constant weight factors are chosen and their values were taken from the last iteration of the PERMAS optimization above that utilizes dynamic weight factors ($k_x = 0.387$, $k_y = 0.346$, $k_z = 0.267$) and already came very close to a castable structure. Hereby, it is assumed that these weight factors satisfy an equal consideration of all load cases (Eq. 2.42) fairly well.

This time, reasonable results could be achieved both with and without the demoulding definition. As time is a limited factor for this thesis, the topology optimization tasks are finished with OptiStruct.
Different variants with a mass of 100 %, 90 %, 80 % and 70 % of the reference mass m_0 are computed and depicted below. The stepwise removal of material demonstrates where the essential material that mainly contributes to the stiffness of the structure is located.

The 100 % m_0 variant shows the cross-connection of the upper two support that is still indicated for 90 % m_0 but then removed completely. The solver also tends to separate the upper left support to two smaller structures for the 100 % and 70 % variant which increases the bending stiffness for a moment applied in this particular plane. The basic shapes of the other two connections, however, do not vary significantly as the mass constraint decreases. Their shape just becomes more slender.



(a) *Variant m100 (100 % m_0)*

(b) *Variant m90 (90 % m_0)*

(c) *Variant m80 (80 % m_0)*

(d) *Variant m70 (70 % m_0)*

Figure 4.7: *Topology optimization variants with 100 % m_0, 90 % m_0, 80 % m_0, 70 % m_0 based on weighted uni-axial loads (x, y, z) with $k_x = 0.387$, $k_y = 0.346$, $k_z = 0.267$*

The evaluation criterion for these variants is their stiffness in comparison to the existing engine mount bracket that is measured by compliance, i.e. global elastic strain energy. For a more detailed insight of the new variants, compliance is first shown for uni-axial loads (x, y, z) and then those values are combined using the dynamic weight factors.

Variant	Relative mass	C_x [mJ]	C_y [mJ]	C_z [mJ]	C_{xyz} [mJ]
Current design (reference)	100 %	49.36	19.10	13.81	**29.40**
Optimized design *m100*	100 %	32.52	16.65	9.19	**20.80**
Optimized design *m90*	90 %	36.30	19.59	9.56	**23.38**
Optimized design *m80*	80 %	40.42	20.59	11.84	**25.93**
Optimized design *m70*	70 %	47.83	20.08	17.52	**30.14**

Table 4.5: Comparison of compliance for the existing engine mount bracket and topology optimization variants (weight factors for the combined compliance C_{xyz}: $k_x = 0.387$, $k_y = 0.346$, $k_z = 0.267$)

The approach for topology optimization that accounts for all load directions x, y and z equally leads to a more balanced design regarding stiffness compared to the old engine mount. The latter one is quite soft in x-direction compared to y- and z-direction. All topology optimizations have a much better relation of the uni-axial stiffnesses and an overall lower compliance to weight ratio.

In conclusion, every weight variant is stiffer in x-direction than the reference design. The originally very low stiffness in y-direction is not reached for $90\,\%m_0$ and below. However, the difference is between 2.6 % and 7.8 % which is considerably small. Regarding stiffness in z-direction, only the 70 % variant loses the comparison while all other variants show similar advantages as for the x-case.
Finally, only the 70 % variant has a smaller weighted compliance C_{xyz} than the current engine mount. The 80 % variant still has a compliance advantage of more than 10 % but it is assumed that the derivation of a smooth CAD model would "cost" stiffness. Thus, no 75 % variant is calculated afterwards, the variant of choice for all consecutive analyses is the optimized design *m80*.

4.2.2 Derivation of a CAD model from topology optimization

The derivation of a CAD model in CATIA from the topology optimization *m80* can be seen in Fig. 4.8 and in more detail in App. C.7. The result is a smooth and castable geometry that has an even further reduced weight with $m = 0.77\,m_0$.

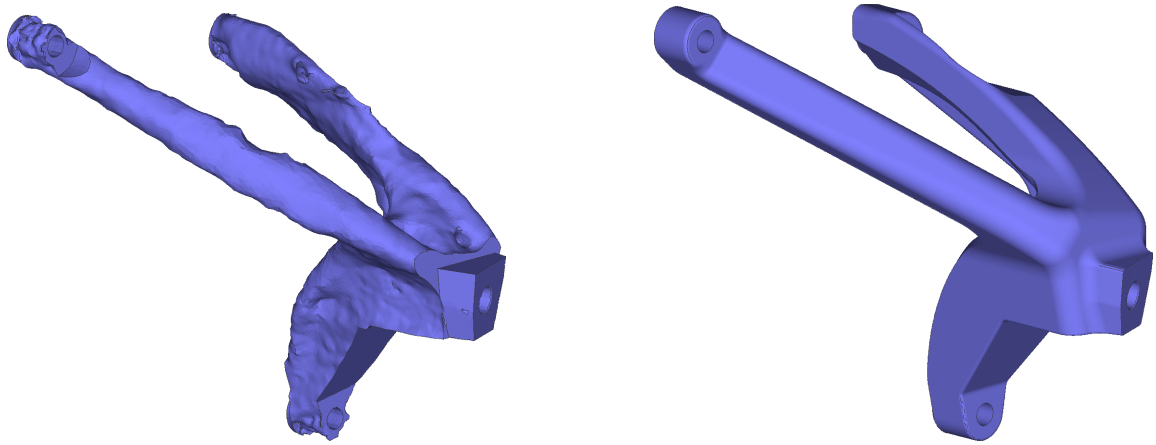


Figure 4.8: *Comparison of the topology optimization and CAD model*

In order to investigate the effect of smoothing of the model, a linear-elastic analysis is performed for the new design. The compliance value in x-direction, C_x, has improved while C_y and C_z now shower worse (higher) values. A full overview of results is summarized in Tab. 4.6.

Variant	Relative mass	C_x [mJ]	C_y [mJ]	C_z [mJ]	C_{xyz} [mJ]
Current design (reference)	100 %	49.36	19.10	13.81	**29.40**
Optimized design *m80*	80 %	40.42	20.59	11.84	**25.93**
Smooth *m80* design: *topo*	77 %	36.96	22.35	12.77	**25.45**

Table 4.6: Comparison of existing engine mount bracket and derived CAD model (weight factors for combined C_{xyz}: $k_x = 0.387$, $k_y = 0.346$, $k_z = 0.267$)

Surprisingly, the total compliance C_{xyz} of the new model is even slightly than before although the weight is less. It is assumed that locally existing irregularities on the surface of the topology optimization do not contribute as much to the stiffness of the structure as to its weight.

4.2.3 Extension from an engine mount bracket to a half engine model

The extension to a half engine model which is justified in Ch. 3.2.4 means that the number of degrees of freedom is increased significantly, see Tab 4.7. In order to keep the size of the problem as small as possible, all components attached to the engine mount only have fine meshes in the connection areas, elsewhere it is quite coarse. Therefore, all added components together only have approximately as much nodes as the engine mount bracket itself although they occupy much more space, see Tab. 4.7.

Component	n_{nodes}	n_{elements}	n_{dof}
Engine mount bracket	2,529,777	1,759,363	7,589,331
Crank case (right half)	719,534	453,313	2,158,602
Upper part of oil pan (right half)	558,236	351,792	1,674,708
Lower part of oil pan (right half)	51,003	25,115	153,009
Bearing bridges (right half)	70,683	41,295	212,049
Solid core of engine bearing	682,095	473,463	2,046,285
Bolt A	69,742	46,970	209,226
Bolt B	52,719	35,079	158,157
Bolt C	60,406	40,249	181,218
Bolt D	123,553	84,468	370,659
RBE2 elements (for introduction of load)	1	1	3
Spring elements (for connectivity of model)	0	25	25
Sum of all components	4,912,914	3,311,133	14,738,767

Table 4.7: Number of nodes, elements and DOFs of extended half engine model

The half engine model is depicted in Fig. 4.9. For reasons of secrecy, the new components are edited since they are in the development stage.

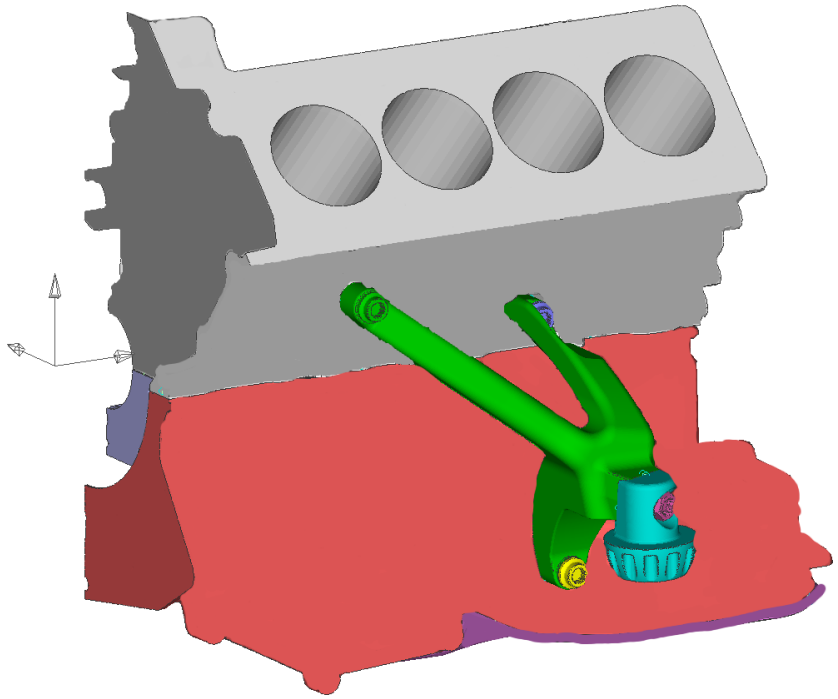


Figure 4.9: *Half engine model (simplified depiction)*

4.2.4 Stress and multi-axial fatigue analysis

First, a stress analysis is performed for load cases which are relevant for operational strength, see Tab. 3.6. The element stress results for the cases "full assembly" ($t = 2$), "full assembly + right corner" ($t = 3$) and ($t = 4$) are read in FEMFAT and the operational strength analysis is performed. The safety factors SF_A over the component before the optimization are shown in Fig. 4.10. Red and orange coloured areas are critical, i.e. $SF_A < SF_{A,\min}$.
Two critical spots can be determined: the first one is located between the lower through hole and the connection of the rubber engine-body mount and has the lowest safety value $SF_{A,\text{critical1}} = 0.79 \cdot SF_{A,\min}$. The second one is located on the right side of the connection of the rubber engine-body mount and closely fails the safety limit with $SF_{A,\text{critical2}} = 0.92 \cdot SF_{A,\min}$.

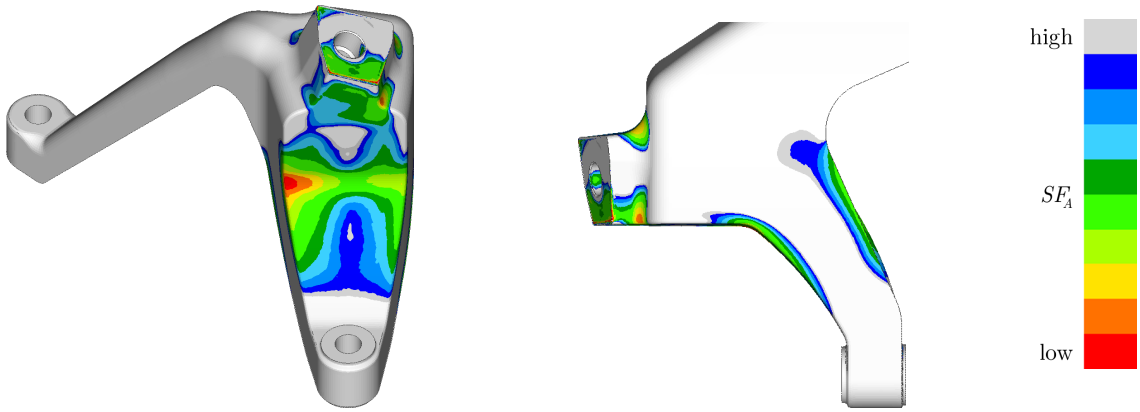


Figure 4.10: *Safety factor SF_A over the component*

Given these results from FEMFAT, a shape modification for both locations has to increase the safety factor for operational strength.

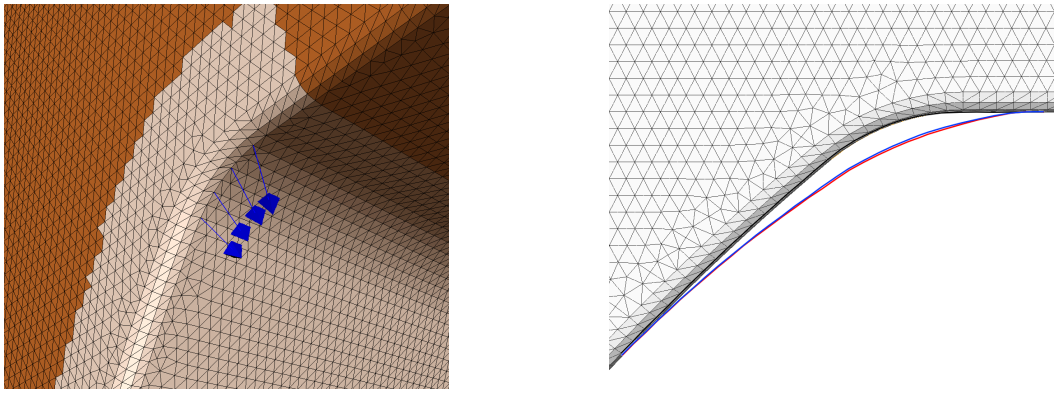
4.2.5 Shape optimization

As mentioned earlier, the goal of this shape optimization is to reduce "amplitude stresses" as shown in the operational strength analysis in FEMFAT. Therefore two different approaches are compared:
In the first approach, simply the von Mises equivalent stress from time step #4, $\sigma_{\text{vM}}(t = 4)$ is selected for the stress constraint. One is aware that this stress state also includes mean stresses that come from the constant assembly loads. The question now is if this stress that includes both mean and amplitude portions is a good measure to reduce the amplitude stress.
Therefore, the "mean stresses" from assembly $\sigma_{\text{vM}}(t = 2)$ are subtracted from $\sigma_{\text{vM}}(t = 4)$ in the second approach and this is taken as a stress constraint. The results in Tab. 4.8 show slightly different modification of the design variables for the two approaches which is also illustrated in Fig. 4.11.

	Relative goal function (weight of design space)	**DV1**	**DV2**	**DV3**	**DV4**	**Relative SF_A**
Before optimization	100 %	0.000	0.000	0.000	0.000	100 %
Shape optimization #1	100.01 %	0.323	0.338	0.437	0.397	126.74 %
Shape optimization #2	100.01 %	0.427	0.314	0.486	0.417	128.93 %

Table 4.8: Final value of goal functions and design variables for different types of stress constraints (shape optimization #1: $\sigma_{\text{vM}}(t = 4) \leq \sigma_{\text{vM,max}}$ / shape optimization #2: $\sigma_{\text{vM}}(t = 4) - \sigma_{\text{vM}}(t = 2) \leq \sigma_{\text{vM,max}}$)

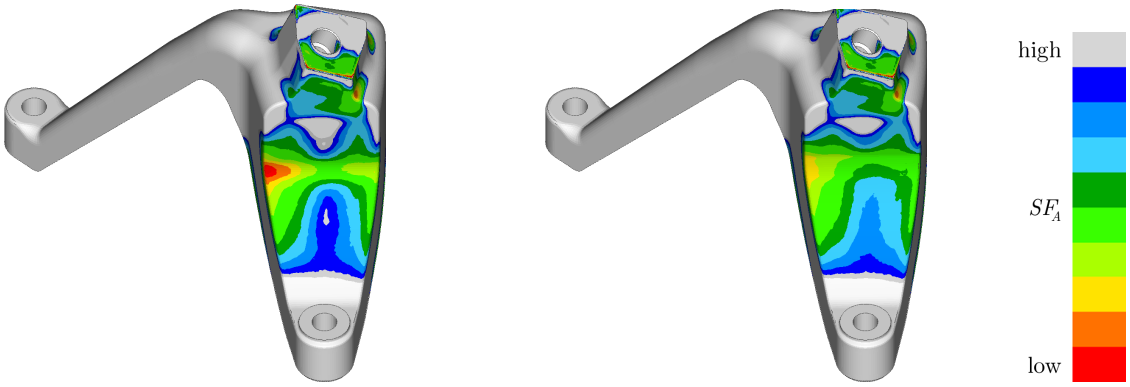
A repeated linear elastic stress analysis with PERMAS and multi-axial fatigue analysis with FEMFAT delivers the new safety factors for the modified structure. As expected, the second approach results in a higher safety value SF_A.



(a) *Design nodes for shape optimization* (b) *Shape modification #1 (blue), #2 (red)*

Figure 4.11: *Shape modification of the most critical spot*

In concrete figures, the safety factor could be increased to $SF_{A,\text{critical1mod}} = 1.02 \cdot SF_{A,\text{min}}$ when the combined element stress was used for the stress constraint (shape optimization #2). In the first approach (shape optimization #1), the safety factor SF_A is only 0.25 % above the safety limit.
This shows that a result combination in order to subtract assembly (mean) stresses is an adequate mean to estimate amplitude stress. The small difference can be explained by the fact that the mean stresses that come from bolt pretensioning are much smaller than amplitude stresses in this example.



(a) SF_A *before shape optimization* (b) SF_A *after shape optimization*

Figure 4.12: *Improvement of the safety factor* SF_A *at most critical spot*

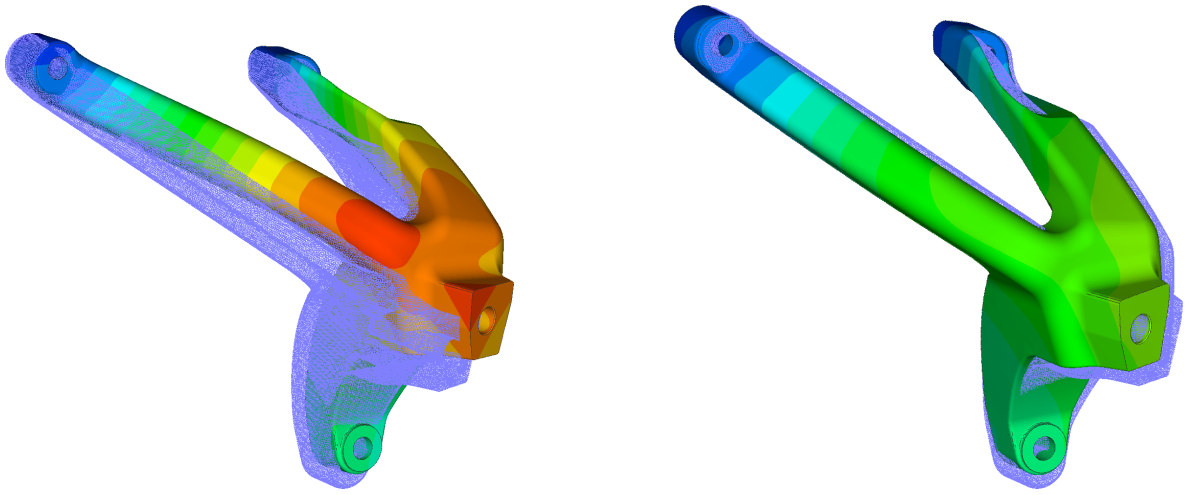
Due to time constraint, the second critical spot could not be optimized. However, this spot was very close to fulfilling the safety limit $SF_{A,\text{min}}$ from the beginning.

4.2.6 Modal analysis

A modal analysis is performed for the half engine model with both the shaped optimized and the old engine mount bracket. All eigenmodes with a frequency between 0 and 1000 Hz are calculated and compared for both variants. Within this range lies the critical eigenfrequency ω_{crit} below which no eigenfrequency should be found. It is noted that ω_{crit} is based on a rough estimate from the NVH department since the drive train is still undergoing development.
A visual check of the eigenmodes in the post-processor reveals that the mode shapes did not change, see Fig. 4.13. However, the different distribution of material leads to lowered eigenfrequencies. The first frequency decreases by approximately $0.07 \cdot \omega_{\text{crit}}$ but is still well above the estimated critical eigenfrequency ω_{crit} for the drive train. A similar drop is found for the second frequency.
Based on the estimate of the NVH department, the optimized engine mount bracket is not critical in terms of vibration. However, this finite element model is still simplified and does not include many of the other

Relative	Half engine model with	
Eigenfrequency	**Existing bracket**	**Optimized bracket**
$\omega_1/\omega_{\text{crit}}$	1.73	1.66
$\omega_2/\omega_{\text{crit}}$	2.09	2.02

Table 4.9: Comparison of relative eigenfrequencies for half engine models with existing and optimized bracket. Reference point: estimated critical eigenfrequency ω_{crit} for drive train.



(a) *First eigenmode* (b) *Second eigenmode*

Figure 4.13: *Eigenmodes of half engine model with optimized bracket (engine components hidden). Blue wireframe indicates undeformed state.*

components of a drive train that could effect the vibration behaviour. Are more detailed analysis as well as vehicle testing could give more reliable results.

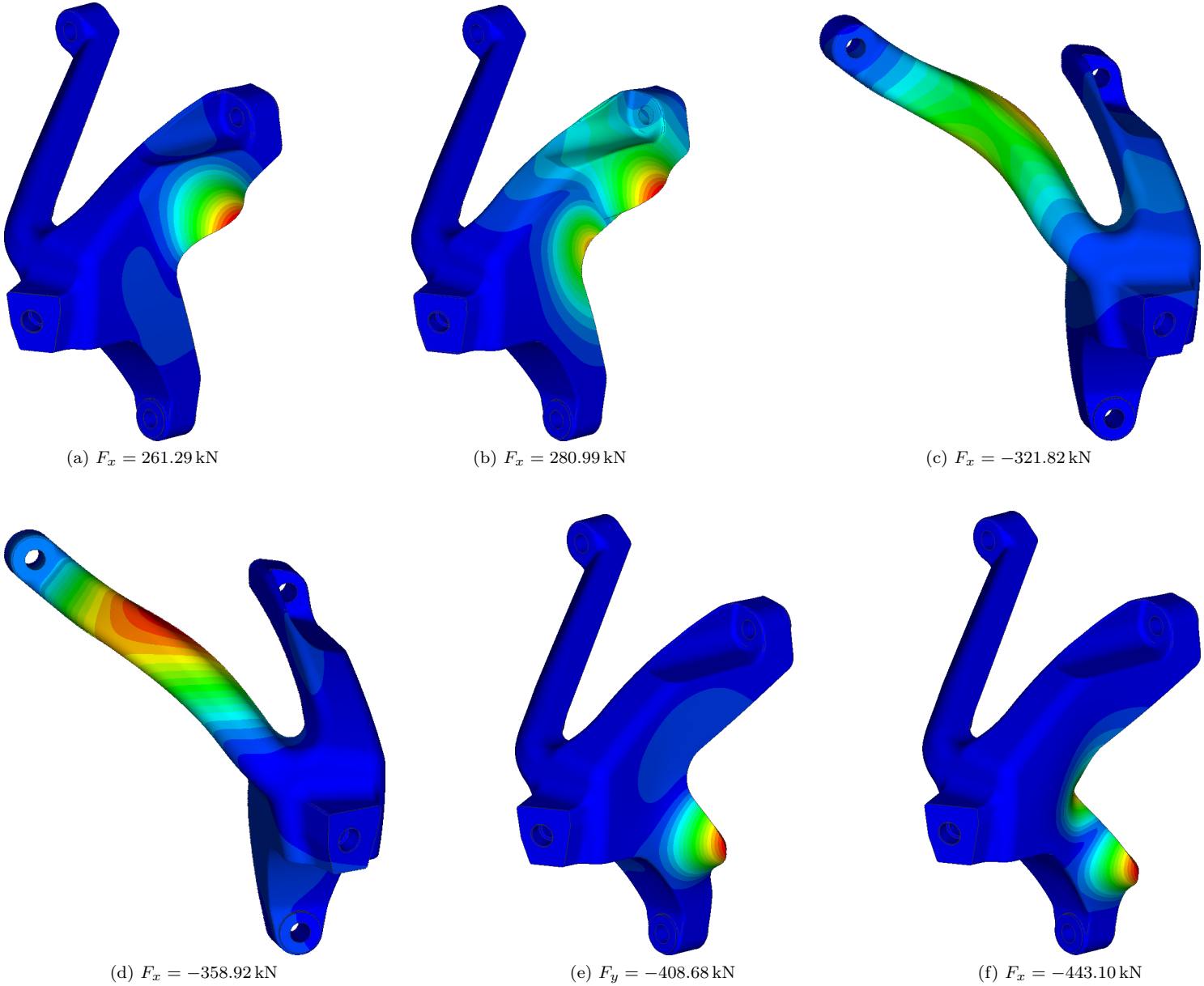
4.2.7 Linear buckling analysis

Last, the linear buckling analysis reveals for which scaled loads the structure collapses as it is subjected to critical compressive stresses. For each loading direction, i.e. x, y and z, a uniform load of 1 kN is applied that is scaled until buckling occurs, whereby the first six buckling modes are calculated.
Not all buckling modes concern the engine bracket itself, some of them occur at other location of the half engine model. Also, none of the first six buckling modes for a z-load affect the engine bracket. The results are summarized in Tab. 4.10 and illustrated in Fig. 4.14.

Load direction	**Mode no.**	**Buckling load**	**Location of buckling**
x-direction	1	$F_x = +261.29\,\text{kN}$	Right upper brace
	2	$F_x = +280.99\,\text{kN}$	Right upper brace
	3	$F_x = -321.82\,\text{kN}$	Left upper brace
	5	$F_x = -358.92\,\text{kN}$	Left upper brace
y-direction	4	$F_y = -408.68\,\text{kN}$	Lower brace
	6	$F_y = -443.10\,\text{kN}$	Lower brace
z-direction	-	none below $\lvert F_z\rvert < 287.23\,\text{kN}$	-

Table 4.10: Buckling loads and location of the engine mount bracket

The magnitude of these buckling loads greatly exceeds any operational load and even crash loads. Thus, there is no risk of buckling.



(a) $F_x = 261.29\,\text{kN}$ (b) $F_x = 280.99\,\text{kN}$ (c) $F_x = -321.82\,\text{kN}$

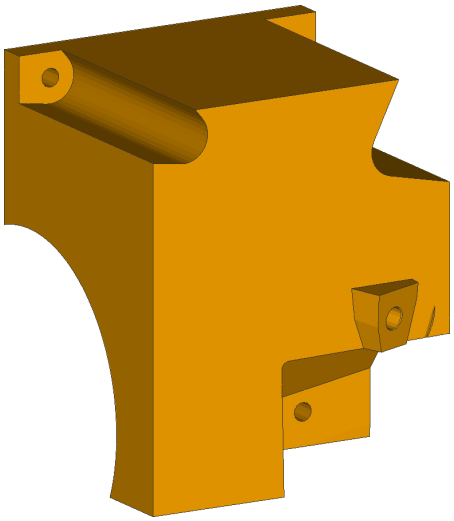
(d) $F_x = -358.92\,\text{kN}$ (e) $F_y = -408.68\,\text{kN}$ (f) $F_x = -443.10\,\text{kN}$

Figure 4.14: *Linear buckling modes for scaled load in x- and y-direction*

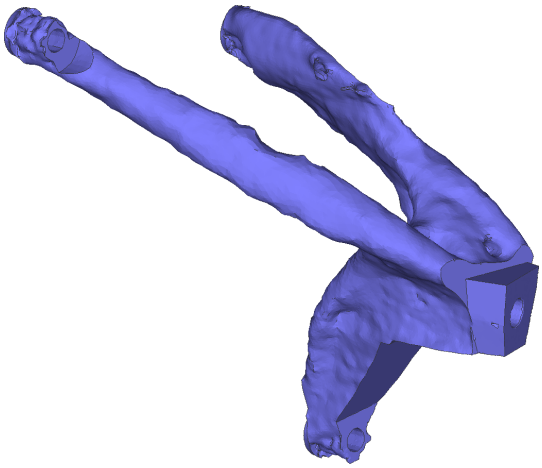
4.3 Overview of all engine mount bracket design stages

The figures below show the complete development progress of the engine mount bracket within this thesis. The following stages were passed to obtain the final design that is depicted in more detail in App. C.11.

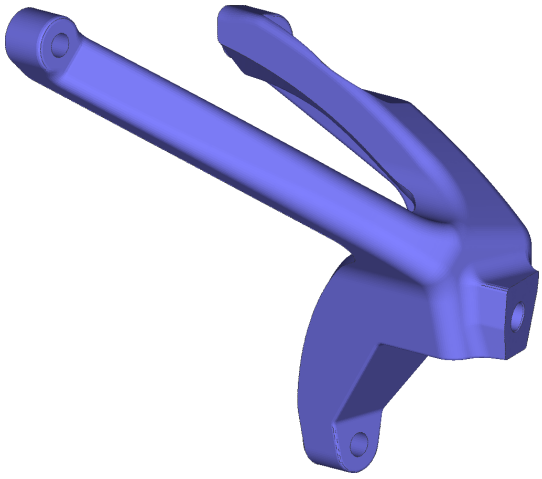
1. Topology optimization (Fig. 4.15a, 4.15b)
2. Derivation of a smooth CAD model (Fig. 4.15c)
3. Shape optimization in conjunction with multi-axial fatigue analysis (Fig. 4.15d)
4. Modal and linear buckling analysis (Fig. 4.15d)



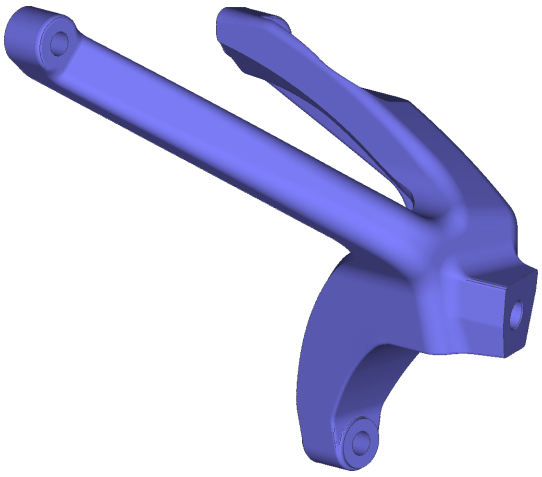
(a) *Design space for topology optimization*



(b) *Topology optimization with* $m = 0.8\,m_0$



(c) *Derivation of smooth CAD model with* $m = 0.77\,m_0$



(d) *Shape-optimized CAD-model with* $m = 0.77\,m_0$

Figure 4.15: *Development stages of the engine mount bracket*

The overall outcome of the optimization performed in this paper is a new engine mount bracket design that **reduces the mass by 23 %** compared to the existing component **without losses in terms of stiffness**.

5 Summary and Conclusion

A series of case studies has been performed to determine appropriate settings for topology and shape optimization models using the finite element software PERMAS. The obtained knowledge was applied when creating a new, weight-optimized engine mount bracket to be used in a Porsche automobile. Apart from optimization calculations, multi-axial fatigue, modal and linear buckling analyses enable further insights in the mechanics of the new part and a comparison with the existing design of the engine mount bracket.

The main results and conclusions are outlined in the next chapters that treat topology, shape optimization and all other analyses separately. A brief outlook rounds off this chapter.

5.1 Topology optimization

The goal of topology optimization is to find an optimal distribution of material within a given design space for a defined load. The iterative assessment of a solution is illustrated for a typical topology optimization example in Fig. B.6.
In automotive industry, the design space is usually provided by the design department and can be reduced prior to optimization by an experienced user who knows which parts of the design space are not likely to be filled with material. This reduction is recommended in cases where the available design space is rather large compared to the sought component since the computational cost of the finite element model increases rapidly. The dependence of analysis time on the number of elements is demonstrated in Tab. 4.3.

The goal function and possible constraints for optimization can be chosen from any mechanical property that is computable. In topology optimization, a common way is to prescribe how much material, i.e. mass, should be filled in the design space. In the sense of optimization, this specified mass constraint is usually less than was used for the previous component. Then, the goal for distributing the material is chosen to result in a maximum stiff structure, of course. This means that the global elastic strain energy, so-called compliance, has to be minimized. This common choice of goal (compliance) and constraint function (mass) is also used for all topology optimizations in this paper, see Eq. 2.44.
Using compliance as a goal function together with a mass constraint, it is shown that a relatively coarse mesh delivers good results for both linear and (reduced) quadratic tetrahedral and hexahedral elements (Fig. B.1– B.4). Therefore, no benefit is expected from quadratic elements and a mesh size that sufficiently approximates the design space geometry and considers all minimal features (e.g. wall thickness etc.) is recommended. For simple geometries that can be meshed with brick elements with little effort, linear hexahedral elements can be used. Otherwise, linear tetrahedral elements deliver good results.

Another very important aspect in topology optimization is the treatment of multiple load cases which takes into consideration that the sought distribution of material must be optimal for all these load cases (Ch. 2.2.1.3). Three different ways for treating multiple load cases in topology optimization have been tested. The first and default method in PERMAS considers only the load case that has the largest contribution to the goal function while all remaining load cases are ignored. Even though this might be reasonable from a mathematical perspective, there is a risk that the final structure is not capable of sustaining load cases that were ignored during the optimization. Therefore, it is recommended to manually assign constant weight factors to the individual loads or to define a mathematical condition such as $k_i C_i = \text{const.}$ that allows for a dynamic weighting. In PERMAS, the latter method requires a Fortran subroutine that has been programmed for that purpose, see App. C.1.3.

In addition, a number of useful settings can be specified in topology optimization which are relevant for manufacturing processes. Those are (cyclic) symmetry, definition of release direction (casting) and minimum member size or wall thickness (Ch. 3.2.2.3). All of them worked fine in PERMAS v14 except the definition of the release direction which was a requirement for the engine mount bracket. Therefore, the topology optimization had to be finished with the software OptiStruct instead which did not encounter these problems. However, this problem has been forwarded to the developers of PERMAS and is currently being addressed.

The solution process itself can be influenced by further parameters that concern the type of topology optimization (Ch. 2.1.7.1), the approximation method (Ch. 2.1.5) and convergence criteria. The type of topology optimization should be chosen to be the SIMP method which is the standard method for regular topology optimization problems nowadays.

Regarding approximation methods for the mathematical optimization problem, all methods offered by PERMAS (Tab. 3.2) except SQP and SQPS show a clear distribution of material and a very similar stiffness for the case problem, see Ch. B.1.2. The calculation time for the case study hereby varies between almost 22 and 32 minutes. However, the individual performance of each method is very dependent on the problem type. Considering the fact that the convex methods GCA (default by PERMAS) and SCP also have the best performance when non-linearities (e.g. contacts) are present, they are to be favoured in the general case. Due to problems during the topology optimization of the engine mount bracket with PERMAS, a final choice between GCA and SCP could not be made.

The default value for tolerance for convergence (of the goal function) is suitable in the general case. If a bad 0/1-distribution of material is found in the post-processing process, this value can be reduced by the user. Also, maximum constraint violations and a tolerance criterion for the design variables, i.e. element filling ratios, can be specified which is rarely necessary.

The final step in topology optimization is the process of deriving a CAD model from optimization result (Ch. 3.2.2.4). The result of a topology optimization calculation are the filling ratios of elements in the design space which define if an element is filled with material or not. However, a clear distribution of filled material and voids can seldom be found and therefore a cut-off limit of the element filling ratio has to be specified. From experience, a cut-off limit between 0.4 and 0.5 will display as much elements as needed to reach the weight constraint that has been specified.

Based on these findings, the performed topology optimization of the engine mount bracket led to a completely new design draft that offers comparable stiffness as the old design but with a weight that could be reduced by 23 %. The topology optimization and the derivation of a CAD model are shown in Fig. C.4 and Fig. C.7, respectively.

5.2 Shape optimization

The goal of shape optimization is to locally modify the shape of a component in order to optimize with respect to a given criterion while fulfilling optional constraints. In structural optimization, a common task is to reduce stresses that can cause material failure which was investigated in this paper.

Given the fact that stresses are evaluated, linear tetrahedral elements are not suitable because of their commonly known bad stress prediction. Thus, quadratic tetrahedral or linear hexahedral elements should be used. The mesh size should be chosen such that the base geometry can be approximated accurately and that a good stress computation is guaranteed. As a rule of thumb, a maximum angle per element of 30 ° over the fillet should be used when using quadratic tetrahedral elements.

For shape optimization, the design space consists only of a restricted set of elements in the vicinity of the location where modification are to be applied, see Fig. 3.6. In this area, nodes are displaced. Therefore, the design space should be chosen large enough to allow a smooth transition to the non-design space and to avoid the common problem of distorted elements as shown in Fig. B.7.

Common features that are modified in a shape optimization are fillets where locally high stresses occur. Here, it is shown that a total of three to four design nodes on the surface of a fillet are sufficient to allow a smooth shape modification (Ch. 4.1.2.1). Furthermore, it is recommended that each design node is assigned to a separate so-called design element all of which have the same design space. The total shape modification then is obtained by superposing the displacements of all design elements.

As in topology optimization, the solution process can be influenced by a number of parameters. First, one of

the local approximation methods for design sensitivity computation given in Tab. 3.3 has to be chosen. LDR as the only method is not capable of providing a converged solution while all other methods deliver shape modification identical to the eye but calculation times differ considerably (Ch. 4.1.2.3).

In numbers, GCA, which is also chosen by the solver when AUTO is selected, takes up almost twice as much time as all other methods without providing better results, see Tab. 4.4. According to a statement from the PERMAS user support, the convex methods GCA and SCP are most capable of dealing with non-linearities such as contact definitions. Given these reasons, the method of choice is SCP which then also yields plausible results for the engine mount bracket.

The default value for tolerance for convergence (of the goal function) is suitable in the general case but can be modified if the convergence is considered to be too fast or slow, respectively. The same applies to the maximum constraint violations and a tolerance criterion for the design variables. Furthermore, the method for design sensitivity computation can be selected. However, it is recommended to leave it set to the default setting AUTO as the solver decides which method is computationally most favourable.

As one of the tasks in this paper was to provide an engine mount bracket design with proven operational strength, increased focus was put on the stress constraint for the shape optimization. For the given problem, the endurance safety factor is mainly influenced by the amplitude stress that is calculated in a multi-axial fatigue analysis in FEMFAT for given stress states form the load history (Tab. 3.6). Since the amplitude stress cannot be calculated in PERMAS directly, it has to be estimated by combining element stress states from different time steps in the load history such that the mean stress "disappears". The used approach is described in Ch. 3.2.6.3 and led to reasonable results.

With this background, a shape optimization of the engine mount bracket was performed. The modification of a transition fillet (Fig. 4.11) that showed a critical endurance safety factor in a multi-axial fatigue analysis, was enlarged such that operational strength could be assured, see Fig. 4.12.

5.3 Concluding analyses

A complete evaluation of a component to be used in a vehicle requires additional modal and linear buckling analyses that could unveil potential causes of failure.

In general, eigenfrequencies of engine components should be as high as possible in order to avoid excitations from the drive drain that are mainly on the region of a couple of hundred hertz. Although it is possible to specify eigenfrequencies for topology optimization, it is found that this type of constraint does not work well with the SIMP method and three-dimensional solid elements, see Fig. C.6. The reasons is that the element mass matrix is linearly dependent on the element filling ratio while the element stiffness matrix is not. The penalization of the element stiffness (usually cubic: x^3) leads to an incorrect eigenfrequency prediction and those models are very susceptible to local modes.

Given these reasons, a modal analysis has to be done for a finite element model that is derived from a topology optimization result and only includes only fully "filled" elements. Naturally, this step should be done for shape optimization in case the vibrational analysis results are completely unacceptable.

It is shown by comparison to the old engine mount bracket that the new design reduceas the first eigenfrequency of the half engine model slightly, see Ch. 4.2.6. However, it is still above an estimated critical value that has been proposed by the NVH department at Porsche.

Last, a linear buckling analysis should be performed if the new design includes long and slender parts that could cause failure when subjected to compressive stresses. Again, this analysis can be done already before shape optimization in order to possibly reveal a critical buckling behaviour at an early stage of development.

The new engine mount bracket design proved to be uncritical in terms of buckling for various load directions, see Ch. 4.2.7. Based on these results, the new design that was proposed in this paper fulfils all functional requirements of an engine component.

5.4 Outlook

Just by the end of this thesis, a new version of PERMAS is being released that offers some new features and improvements over the current version.

A new non-parametric optimization tool is now available which allows a free-form shape optimization of structures in order to minimize stresses by homogenization or to minimize weight with a stress limit. This enables easy shape optimizations of free-form geometries.
The topology optimization tool has been revised and now provides solution with clear separation of empty and filled elements (0/1 distribution).

In addition, the all-new PERMAS release v15, enables parallel CPU/GPU computation and general performance-oriented improvements such as a renewed DOF renumbering algorithm that will allow faster solving times and larger models.

6 Recommendations for further Work

In the light of the presented results and conclusions, the following topics of structural optimization could be investigated in further works:

- Further research on numerical topology optimization
  - Eigenfrequency as goal or constraint function
  - Anisotropic material in topology optimization
  - Two-dimensional shell elements in topology optimization (thin-walled components)
- Further research on numerical shape optimization
  - CAD-based shape optimization
  - Free-form shape optimization

A Consistent Units for FEM

The units of physical quantities are not declared in the PERMAS input files and therefore consistent dimensional units should be used throughout all analyses. The base units are chosen for coordinates (mm), mass (t), time (s) and temperature (K) whereby units of related physical quantities can be derived. Tab. A.1 gives an overview about the choice of consistent units for this thesis.

Basic units	$[\mathrm{mm}, \mathrm{t}, \mathrm{s}, \mathrm{K}]$
Derived units	$\left[\mathrm{N} = \mathrm{t} \cdot (\mathrm{mm} \cdot \mathrm{s})^{-1}\right]$ $[\mathrm{mJ} = \mathrm{N} \cdot \mathrm{mm}]$
Elastic Young's modulus E	$\left[\mathrm{N} \cdot \mathrm{mm}^{-2}\right]$
Mass density ρ	$\left[\mathrm{t} \cdot \mathrm{mm}^{-3}\right]$
Coordinate x_i	[mm]
Displacement u_i	[mm]
Force F_i	[N]
Reaction force R_i	[N]
Stress σ	$\left[\mathrm{N} \cdot \mathrm{mm}^{-2}\right]$
Energy C	[mJ]

Table A.1: Consistent units for FEM

B Academic Examples

The following appendices include the full documentation of results from the topology and shape optimization examples. Their are left out in the main chapters for the sake of clarity.

B.1 Topology optimization

In this section, further illustration on the topology optimization results for different meshes and element types as well as local approximation methods are given. Also, the convergence progress of a topology optimization is demonstrated by a line-up of several intermediate results.

B.1.1 Mesh size and element types

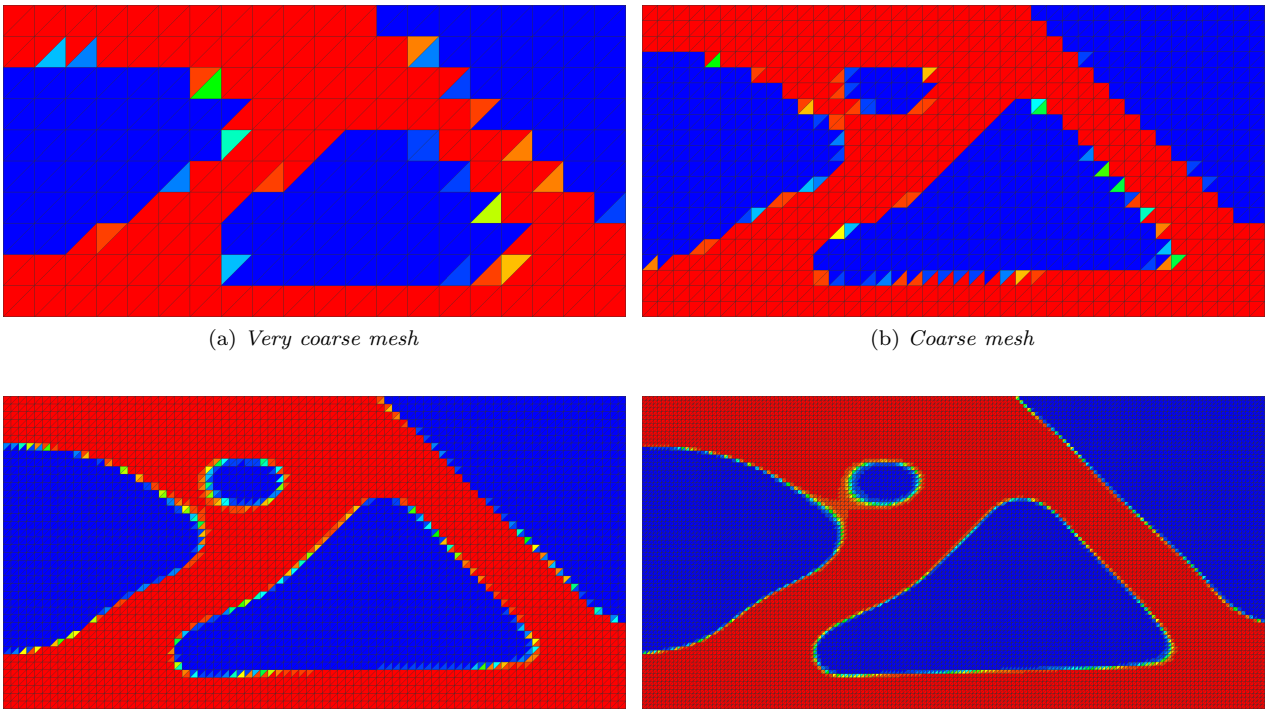
- Results for different TET4 mesh sizes: Fig. B.1
- Results for different TET10 mesh sizes: Fig. B.2
- Results for different HEXE8 mesh sizes: Fig. B.3
- Results for different HEXE20 mesh sizes: Fig. B.4

B.1.2 Local approximation methods

The results for different local approximation methods are shown in Fig. B.5.

B.1.3 Convergence of a topology optimization solution

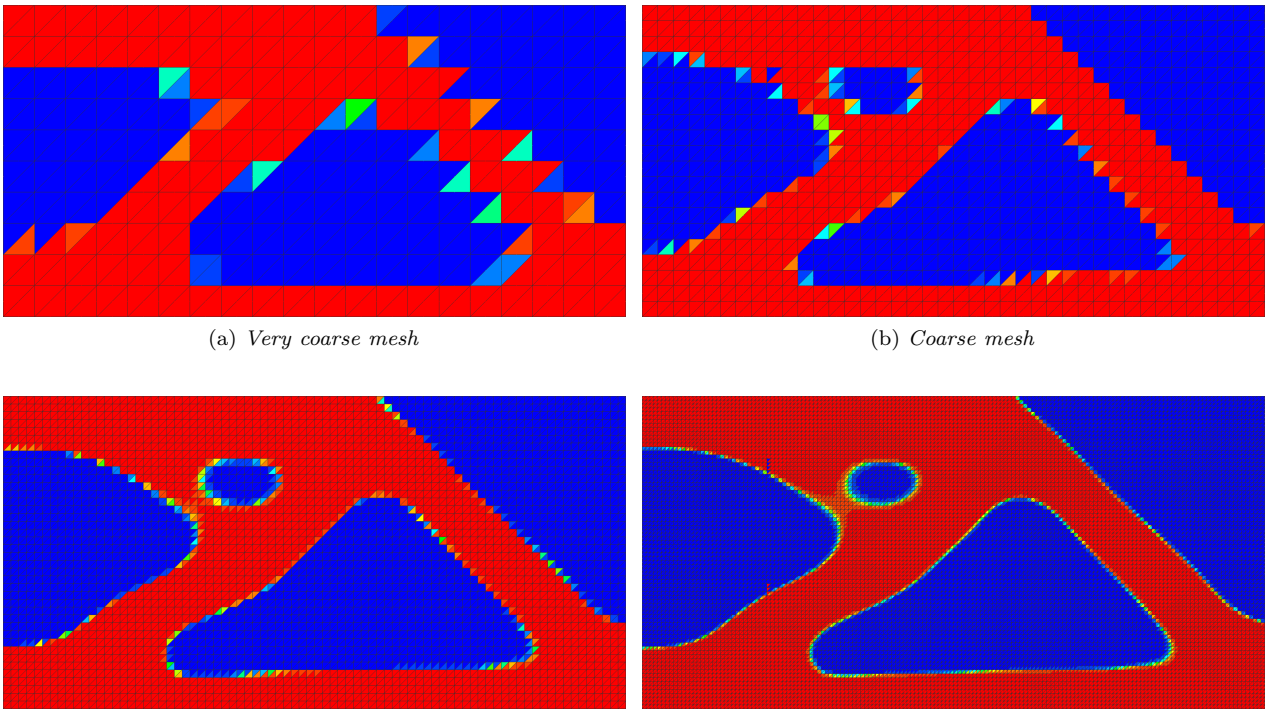
The convergence of a topology optimization is shown in Fig. B.6.



(a) *Very coarse mesh* (b) *Coarse mesh*

(c) *Fine mesh* (d) *Very fine mesh*

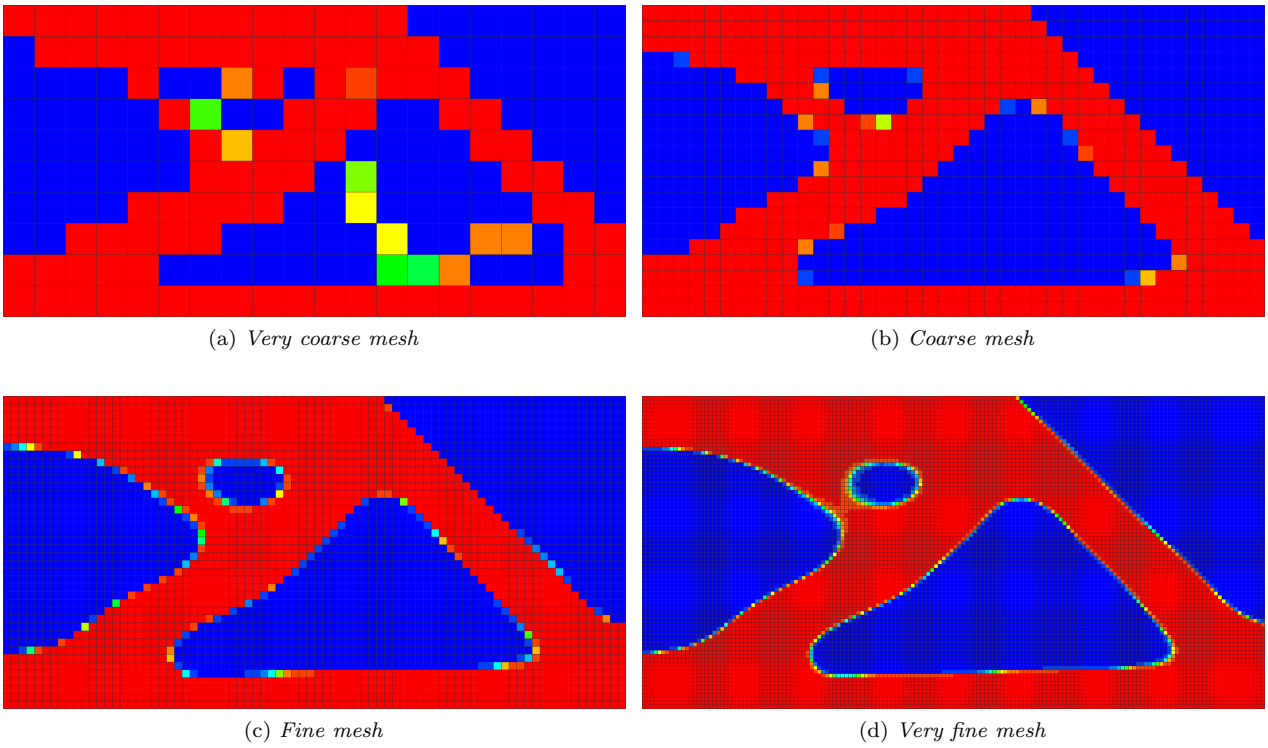
Figure B.1: *Material distribution from topology optimization using local GCA approximation and different* `TET4` *mesh sizes*



(a) *Very coarse mesh* (b) *Coarse mesh*

(c) *Fine mesh* (d) *Very fine mesh*

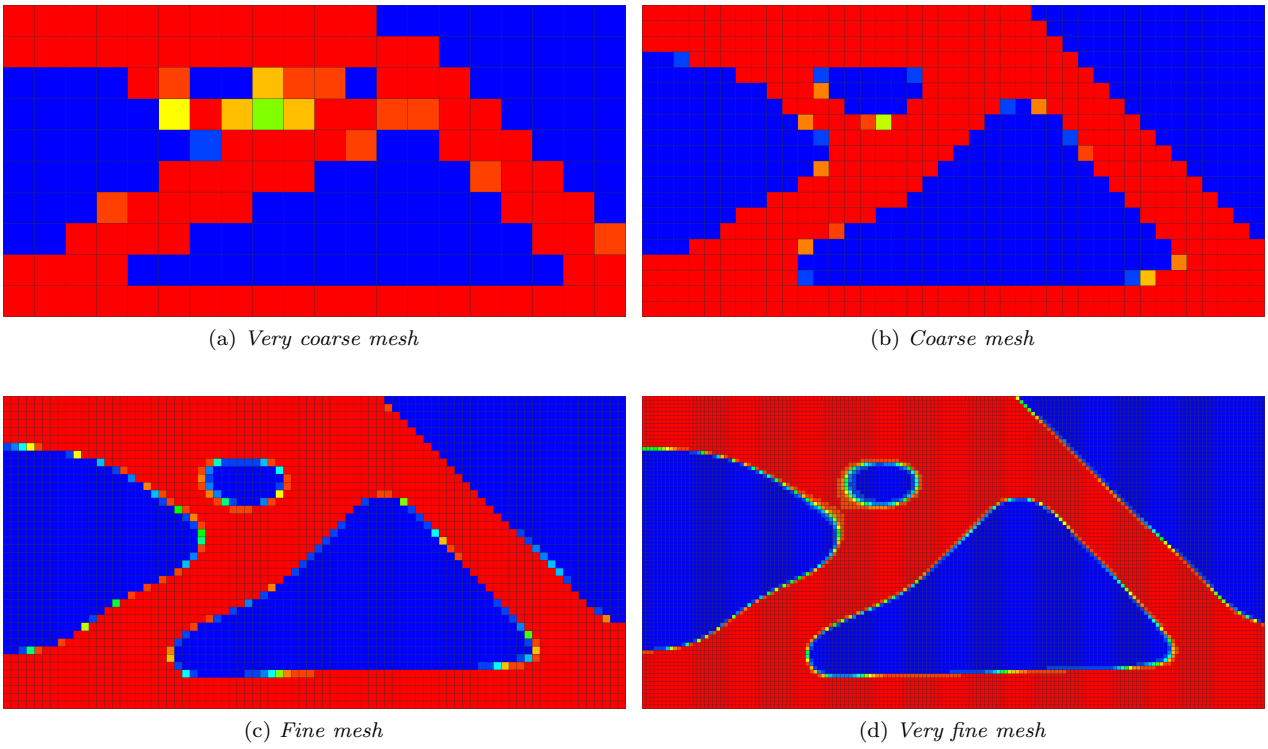
Figure B.2: *Material distribution from topology optimization using local GCA approximation and different* `TET10` *mesh sizes*



(a) *Very coarse mesh* (b) *Coarse mesh*

(c) *Fine mesh* (d) *Very fine mesh*

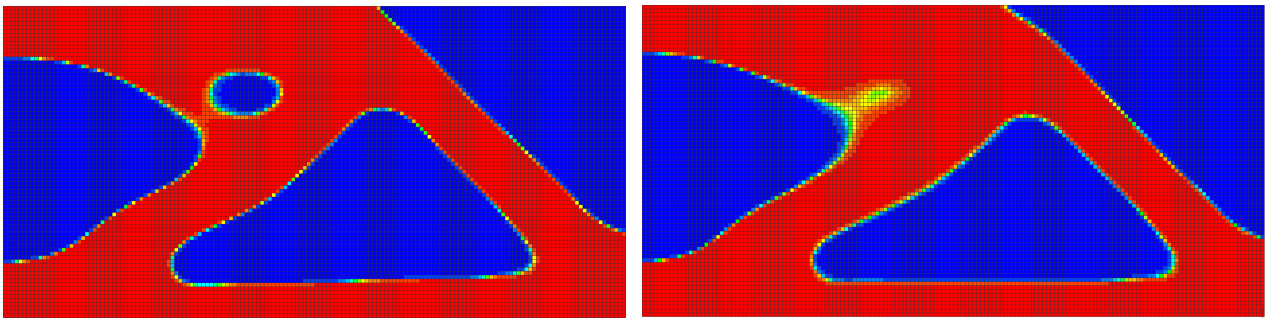
Figure B.3: *Material distribution from topology optimization using local GCA approximation and different* `HEXE8` *mesh sizes*



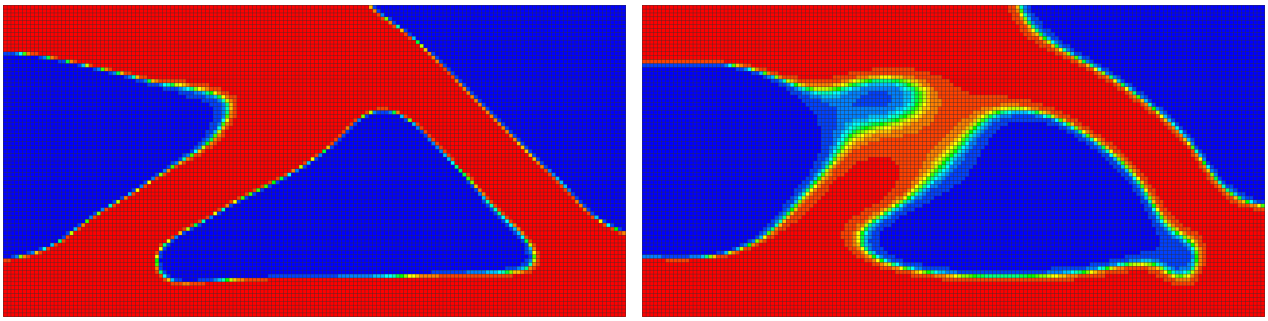
(a) *Very coarse mesh* (b) *Coarse mesh*

(c) *Fine mesh* (d) *Very fine mesh*

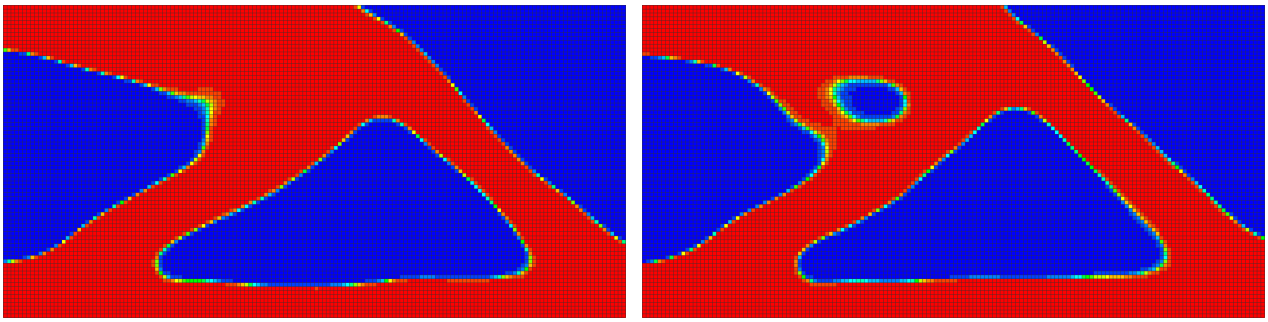
Figure B.4: *Material distribution from topology optimization using local GCA approximation and different* `HEXE20` *mesh sizes*



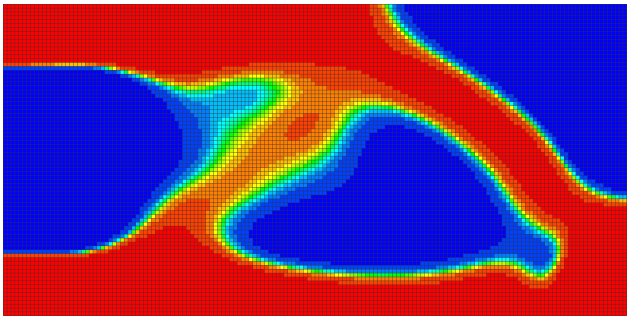
(a) *GCA* (b) *SCP*



(c) *SLP* (d) *SQP*

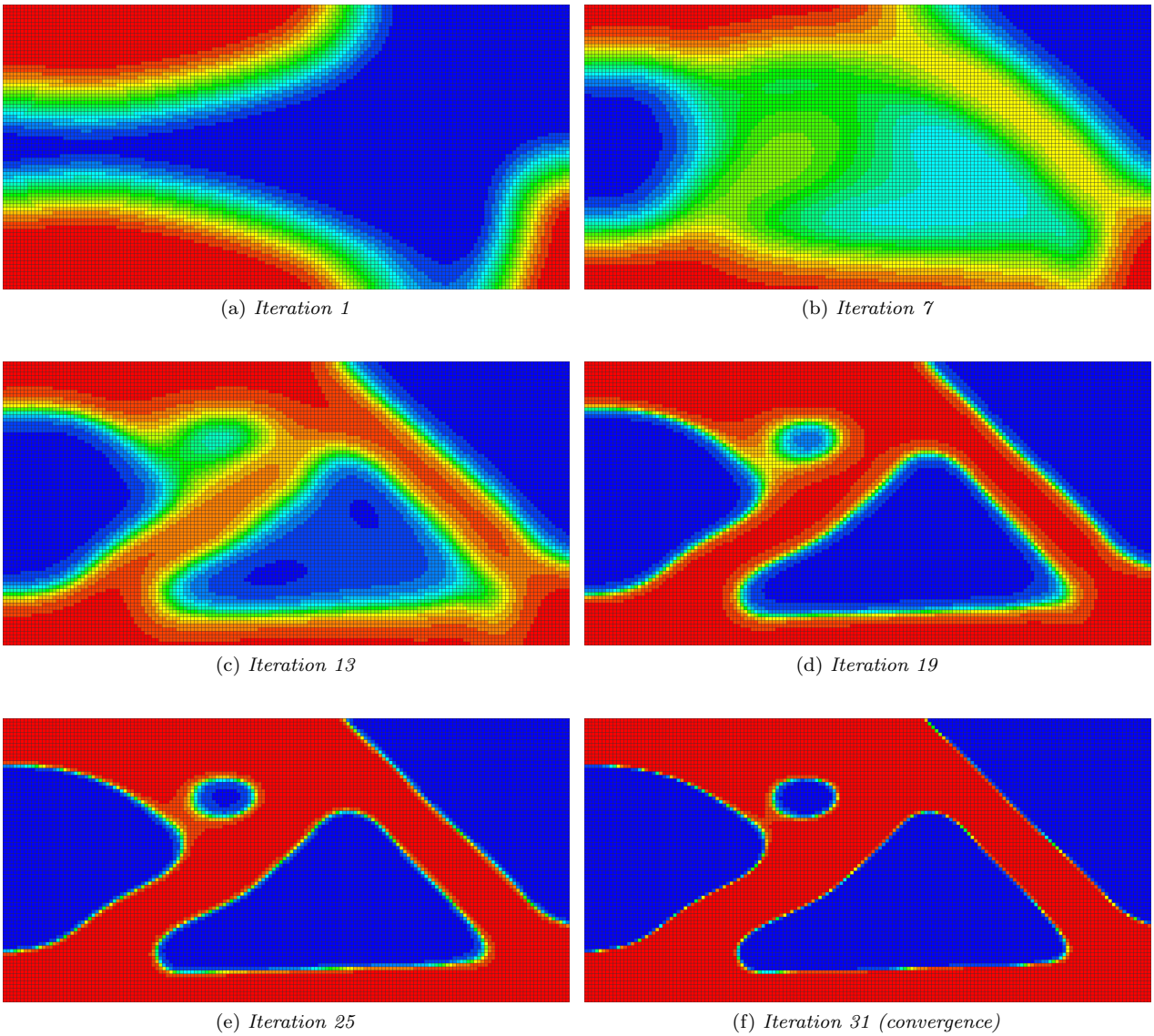


(e) *SCPS* (f) *SLPS*



(g) *SQPS*

Figure B.5: *Material distribution from topology optimization using different solution methods with a very fine* `HEXE8` *mesh*



(a) *Iteration 1*

(b) *Iteration 7*

(c) *Iteration 13*

(d) *Iteration 19*

(e) *Iteration 25*

(f) *Iteration 31 (convergence)*

Figure B.6: *Convergence of a solution using a very fine* `HEXE8` *mesh and the GCA approximation method*

B.2 Shape optimization

In this section, further illustration on the shape optimization results for different number of design variables, meshes and element types as well as local approximation methods are given. Also, the common problem of distorted meshes in shape optimization is illustrated.

B.2.1 Common problems of shape optimization

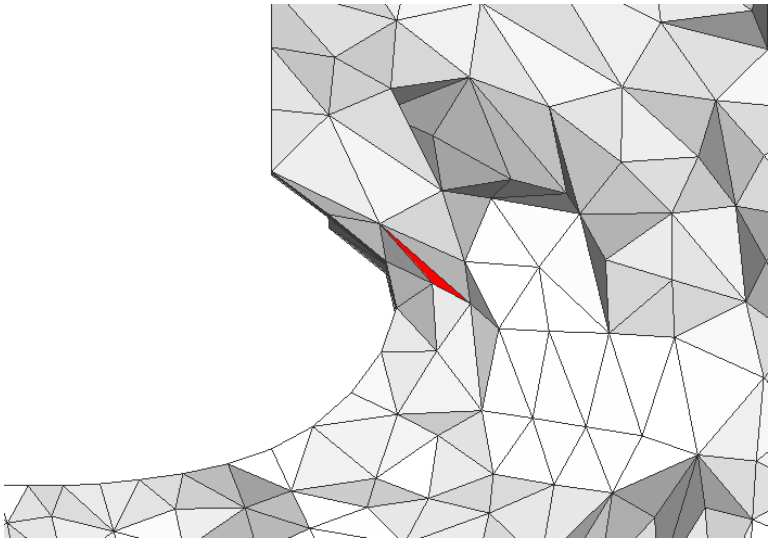


Figure B.7: *Bad shaped element (red) due to large shape modifications*

B.2.2 Number of design variables

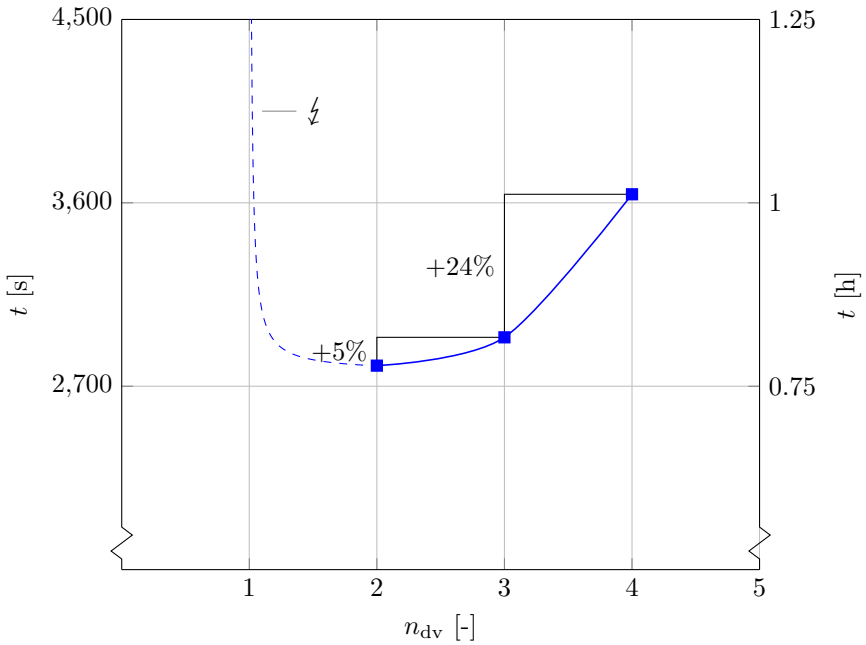
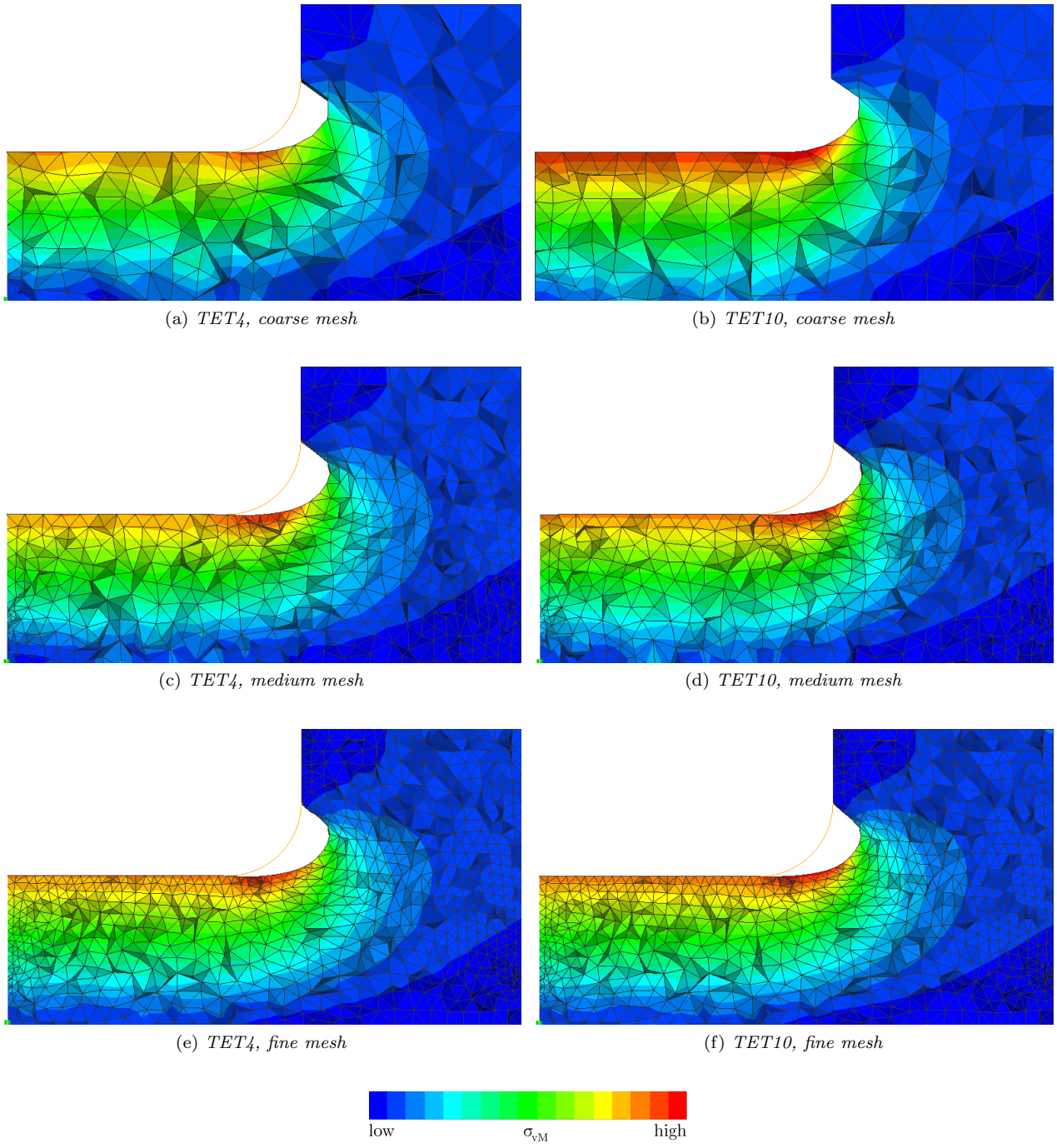


Figure B.8: *Calculation time t vs. number of design variables* n_{dv} *(fine* `HEXE8` *mesh, GCA*

B.2.3 Mesh size and element types

The results for different mesh sizes using TET4 and TET10 elements are shown in Fig. B.9. Analogous to this, results of HEXE8 and HEXE20 elements are illustrated in Fig. B.10.



(a) *TET4, coarse mesh*

(b) *TET10, coarse mesh*

(c) *TET4, medium mesh*

(d) *TET10, medium mesh*

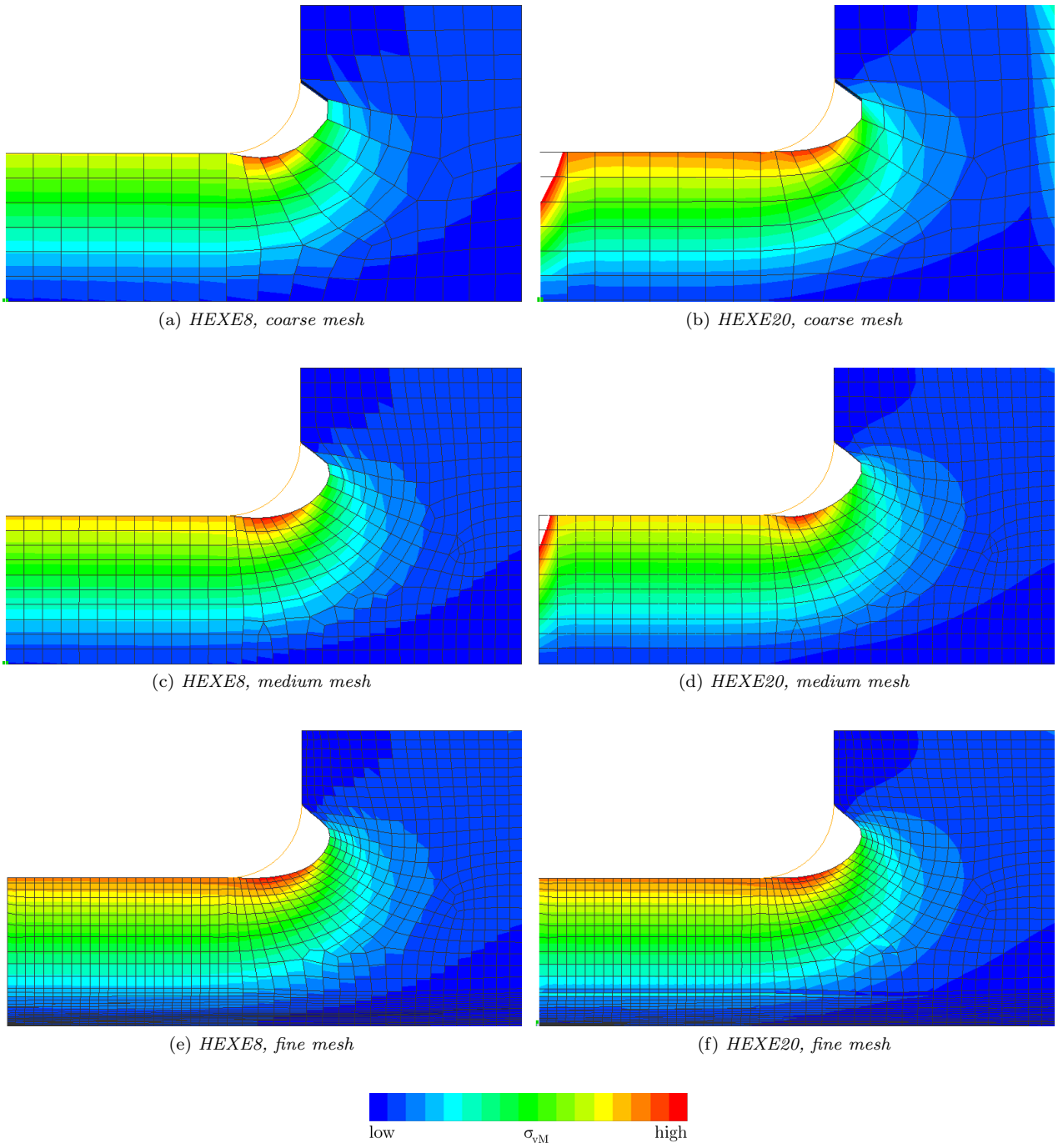
(e) *TET4, fine mesh*

(f) *TET10, fine mesh*

Figure B.9: *Shape modification and nodal point stresses from shape optimization using different mesh sizes with tetrahedral elements*

B.2.4 Local approximation methods

The shape optimization results for local approximation methods are depicted in Fig. B.11.



(a) *HEXE8, coarse mesh*

(b) *HEXE20, coarse mesh*

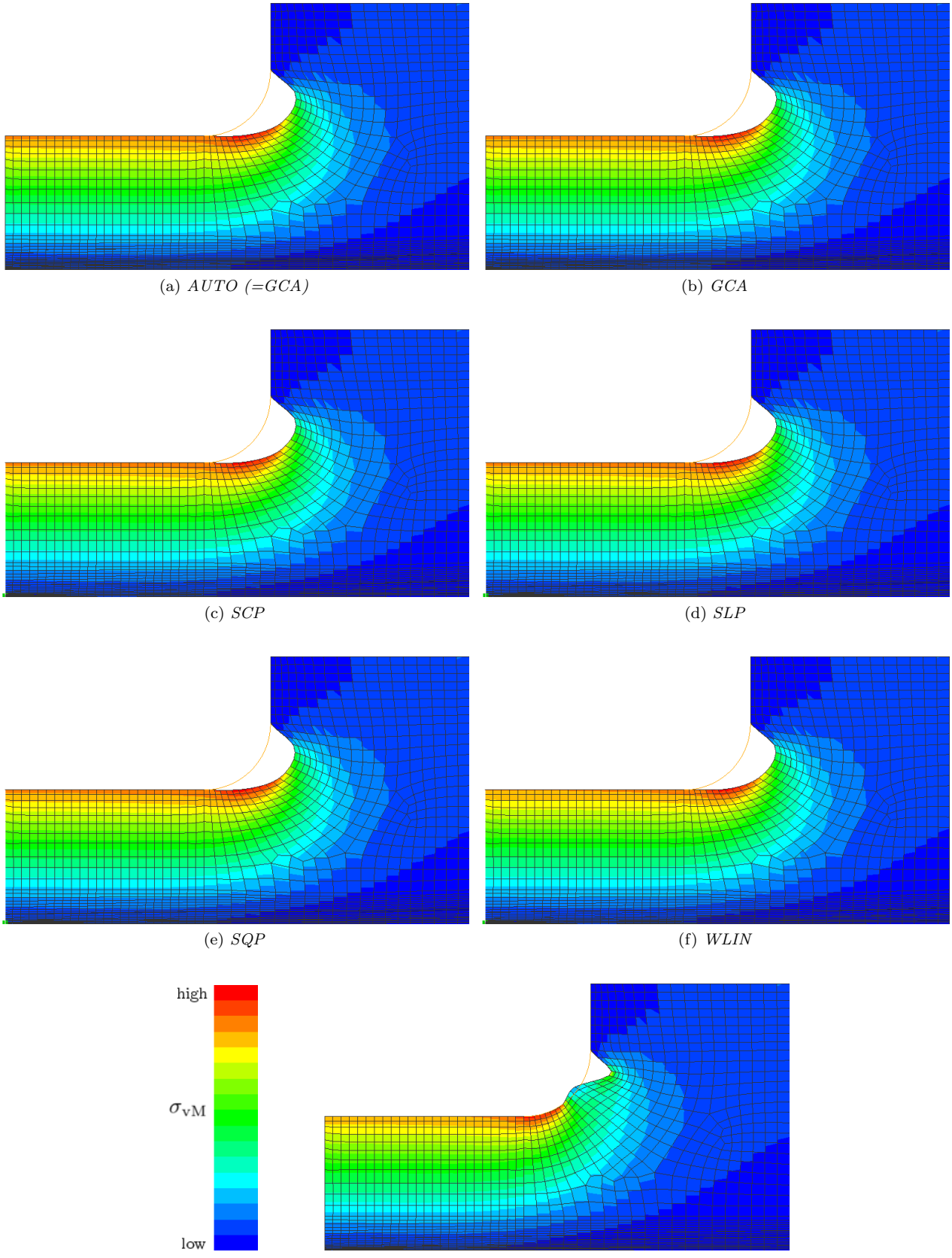
(c) *HEXE8, medium mesh*

(d) *HEXE20, medium mesh*

(e) *HEXE8, fine mesh*

(f) *HEXE20, fine mesh*

Figure B.10: *Shape modification and nodal point stresses from shape optimization using different mesh sizes with hexahedral elements*



(a) *AUTO (=GCA)*

(b) *GCA*

(c) *SCP*

(d) *SLP*

(e) *SQP*

(f) *WLIN*

(h) *LDR*

Figure B.11: *Shape modification and nodal point stresses from shape optimization using different local approximation methods with a fine* `HEXE8` *mesh*

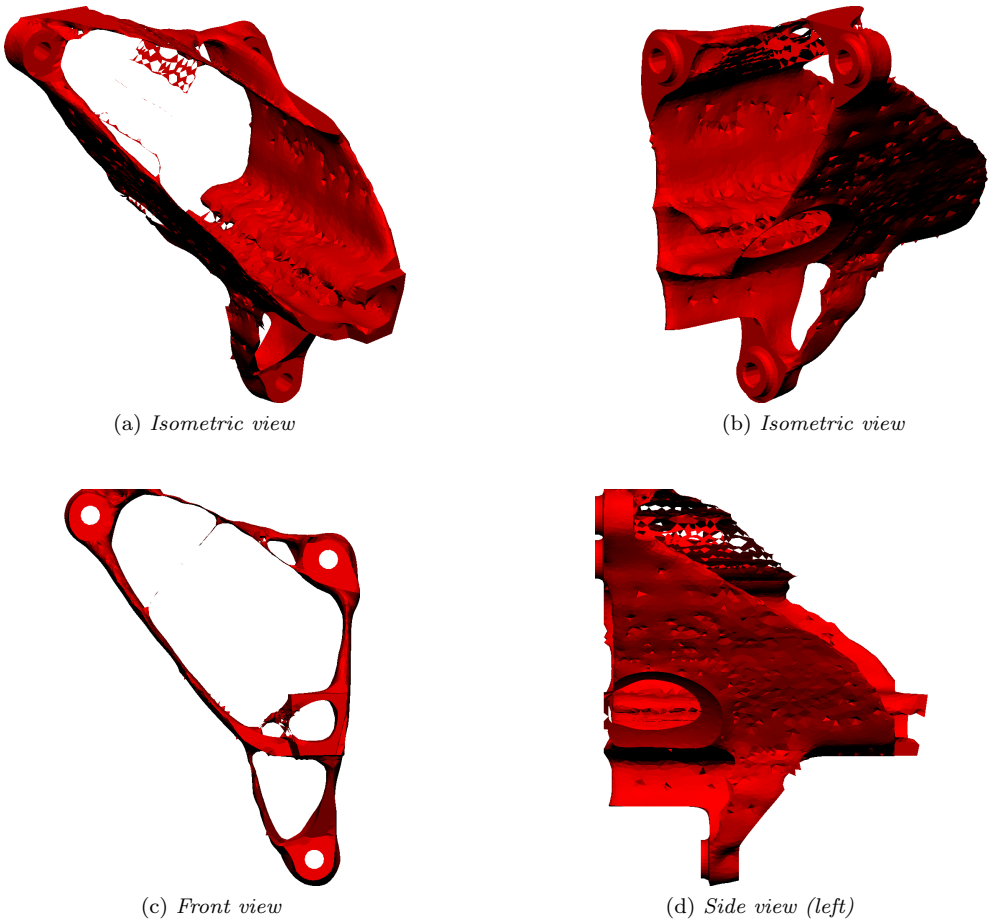
C Structural Optimization of an Engine Mount Bracket

The following appendices provide additional illustration and documentation on topology and shape optimization as well as the multi-axial fatigue analysis that is done in the structural optimization process of an engine mount bracket.

C.1 Topology optimization

In this section, the problems that encountered with PERMAS when specifying a demoulding direction are shown. Additionally, a complete documentation of all OptiStruct results is given. Furthermore, a Fortran subroutine for dynamic weighting factors in PERMAS is appended, followed by results from a topology optimization using eigenfrequency constraints. Last, a detailed comparison of the weight-optimized topology optimization and the derived CAD model is outlined.

C.1.1 PERMAS problems in connection with a specified demoulding direction



(a) *Isometric view*

(b) *Isometric view*

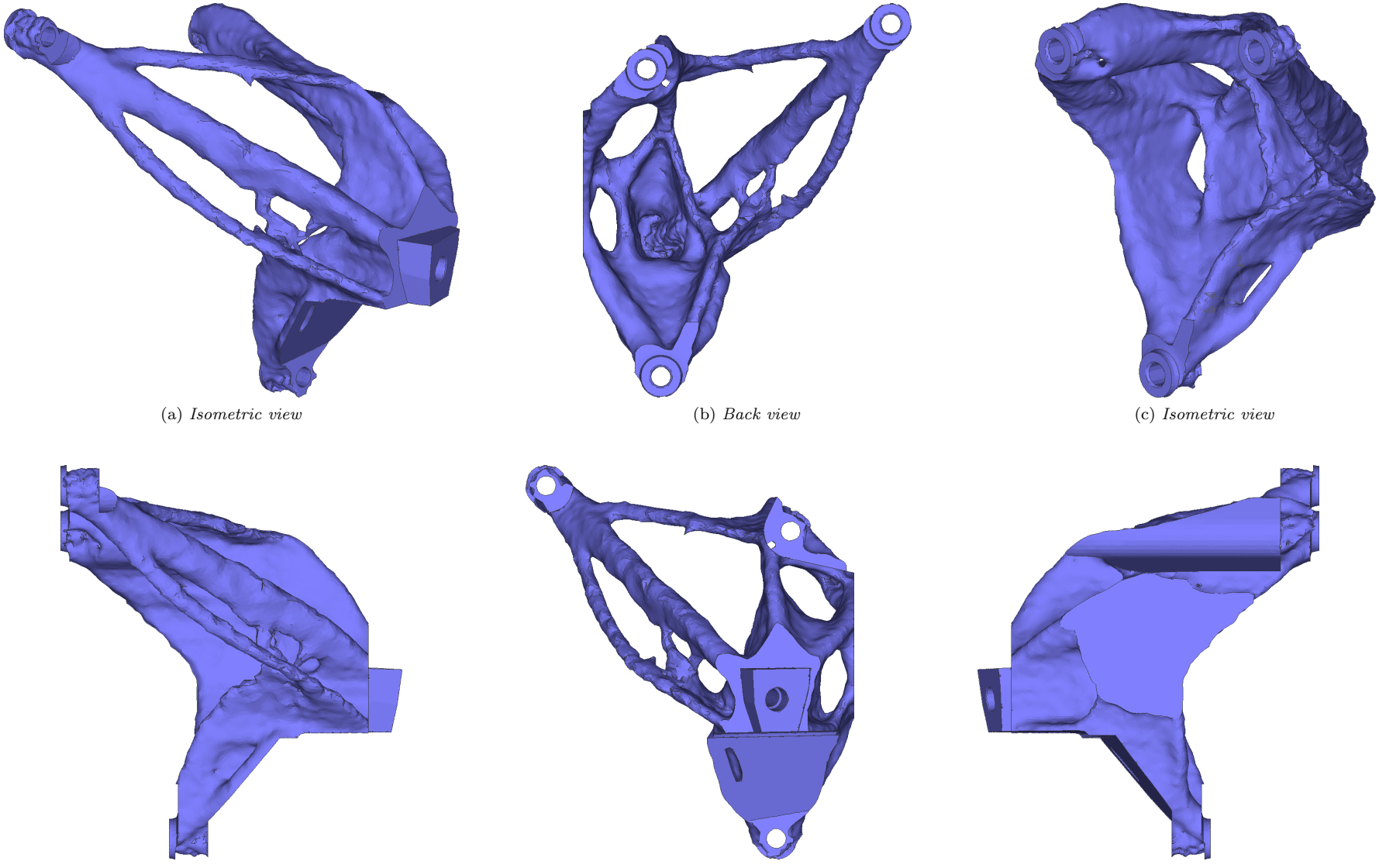
(c) *Front view*

(d) *Side view (left)*

Figure C.1: *Numerical problems encountered with activated two-sided demoulding direction in PERMAS*

C.1.2 OptiStruct results

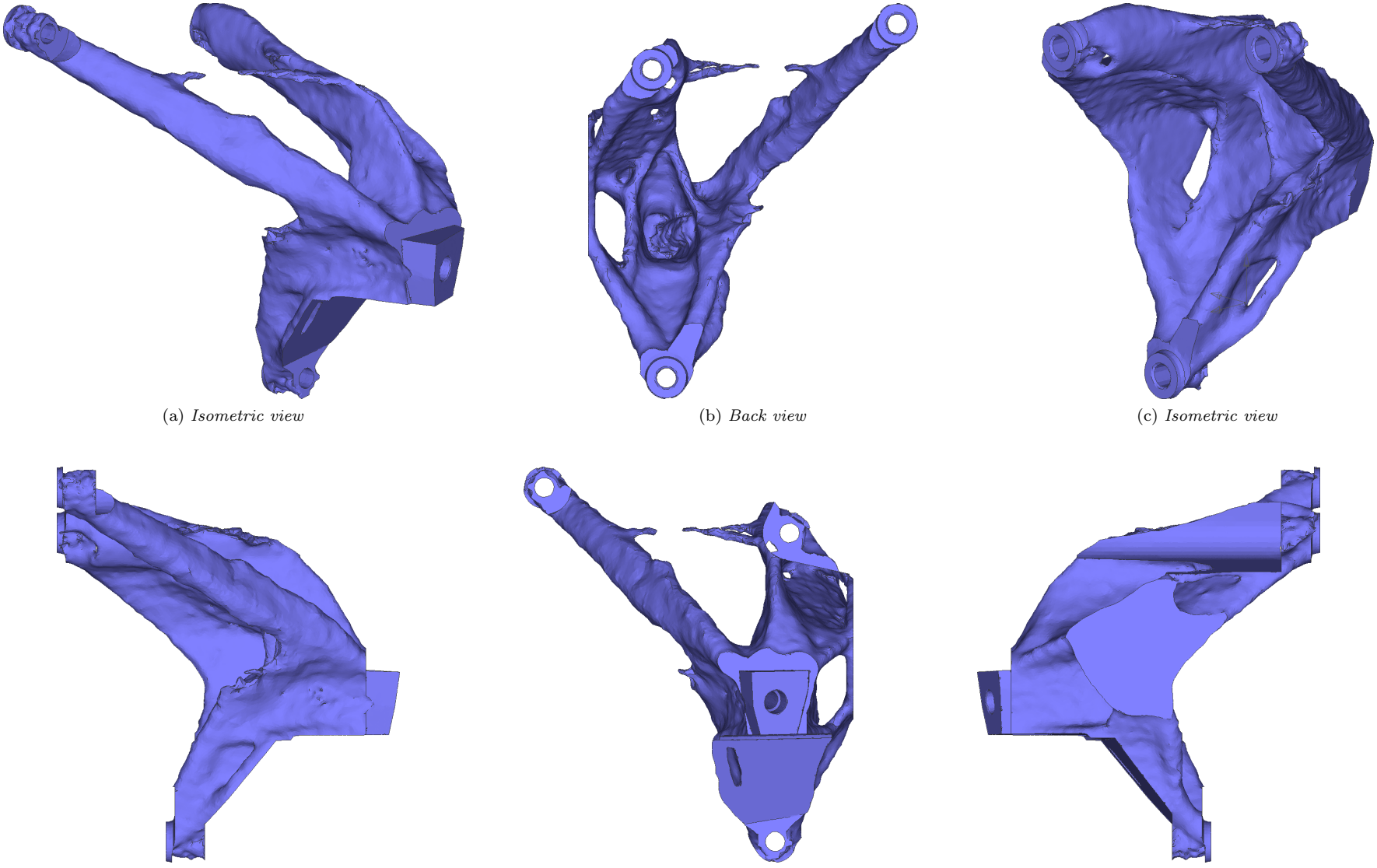
The following pages depict the weight-optimized engine mount bracket variants *m100* (Fig. C.2), *m90* (Fig. C.3), *m80* (Fig. C.4) and *m70* (Fig. C.5).



(a) *Isometric view* (b) *Back view* (c) *Isometric view*

(d) *Side view (left)* (e) *Front view* (f) *Side view (right)*

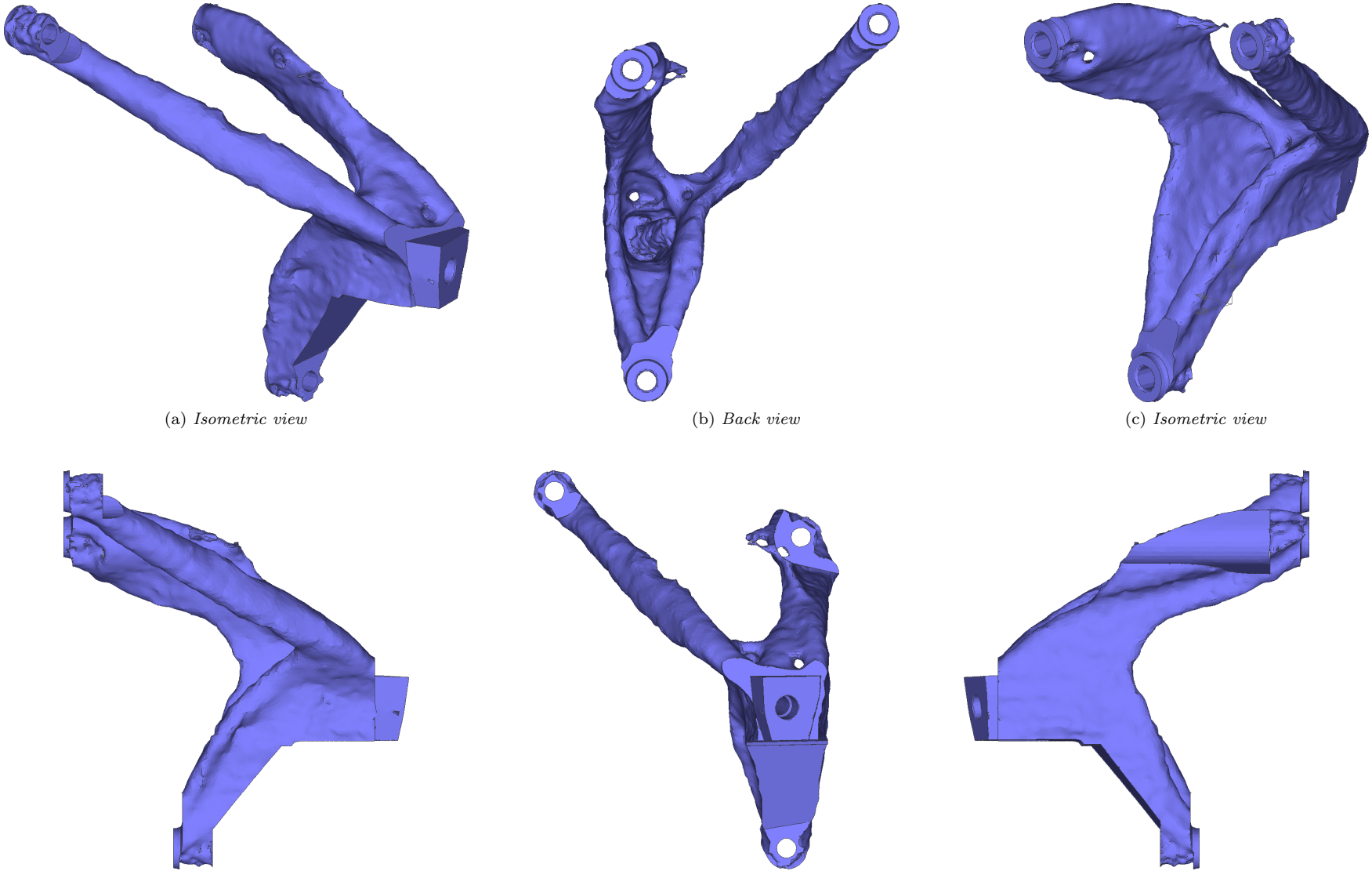
Figure C.2: *Topology optimization for an optimal stiff design with a mass constraint* $m \leq 1.0\, m_0$



(a) *Isometric view* (b) *Back view* (c) *Isometric view*

(d) *Side view (left)* (e) *Front view* (f) *Side view (right)*

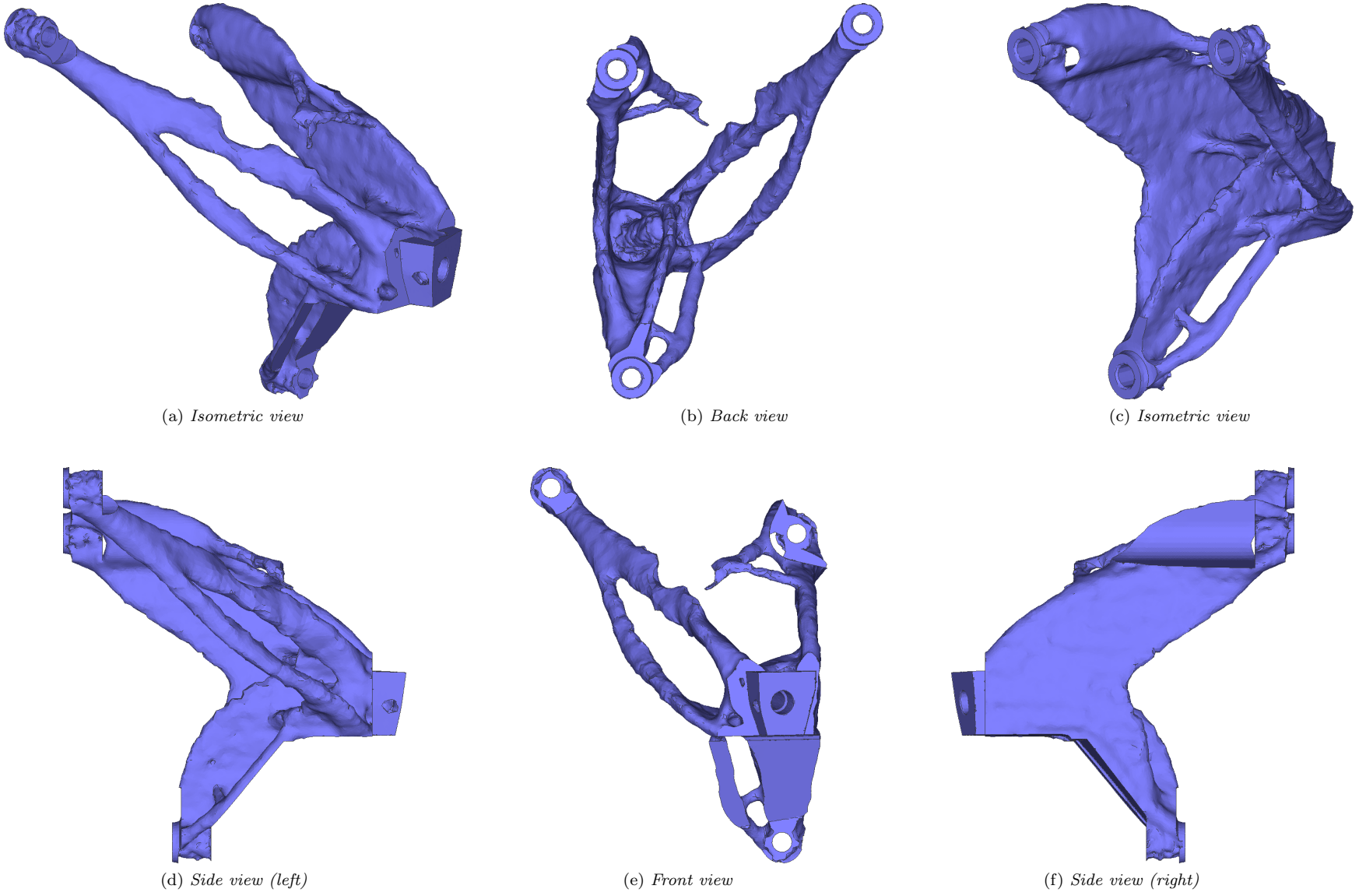
Figure C.3: *Topology optimization for an optimal stiff design with a mass constraint* $m \leq 0.9\, m_0$



(a) *Isometric view* (b) *Back view* (c) *Isometric view*

(d) *Side view (left)* (e) *Front view* (f) *Side view (right)*

Figure C.4: *Topology optimization for an optimal stiff design with a mass constraint* $m \leq 0.8\, m_0$



(a) *Isometric view* (b) *Back view* (c) *Isometric view*

(d) *Side view (left)* (e) *Front view* (f) *Side view (right)*

Figure C.5: *Topology optimization for an optimal stiff design with a mass constraint* $m \leq 0.7\, m_0$

C.1.3 Fortran subroutine for weighted input of goal function

The following code is a Fortran subroutine to be used with PERMAS in order to allow a user-defined goal function. Basically, any mathematical expression can be implemented in the code whereby the partial derivatives of the goal function for design sensitivity analysis also have to be supplied.

```
      SUBROUTINE FNUSRD ( FID, NOC, DATA, X, F, FD, FDD)
      IMPLICIT NONE
      INTEGER    FID, NOC(*)
      REAL       DATA(*), X(*), F, FD(*), FDD(*)
C-------------------------------------------------------------------------------
C PERMAS TOPO: Objective Function
C f ( x1 , x2 , x3 ) = ( x1^2 + x2^2 + x3^2 ) / ( x1 + x2 + x3 )
C -------------
C     FID      :  Function ID
C     NOC      :  1: No. coefficients
C              :  2: No. variables
C     DATA     :  Coefficients
C     X        :  Variables
C     F        :  Function value
C     FD       :  First derivatives
C     FDD      :  Second derivatives
C-------------------------------------------------------------------------------
      LOGICAL    FIRST, OK
      INTEGER    IUNIT, I
      REAL       W(3)
      DATA       FIRST /.TRUE./
      DATA       I /0/
      SAVE       FIRST,I, IUNIT
C
      IF (FIRST) THEN
!                      filename    ext
         CALL SAOPEN ('variables','txt', I U N I T)
         WRITE(IUNIT,1000) '    w1    ','    w2    ','    w3    ',
     +   '  COMPL1  ','  COMPL2  ','  COMPL3  ',
     +   '    F     ','dF/dCOMPL1','dF/dCOMPL2','dF/dCOMPL3'
 1000    FORMAT(10(2X,A10,1X))
         FIRST = .FALSE.
      ENDIF
C
      IF (FID .EQ. 1001) THEN
         PRINT *, FID
C
C
C ***********************
C START YOUR INPUT HERE
C ***********************
C
C ---------------------------------------------
C ------ #1: Objective function F (Zielfunktion)
C ---------------------------------------------
         F = ( X(1)**2 + X(2)**2 + X(3)**2 )
     +       / ( X(1) + X(2) + X(3) )
C
C -------------------------------------------------------------
C ------ #2: Partial derivatives (partielle Ableitungen)
C -------------------------------------------------------------
C ------ Partial derivative d(F) / d(X1)
         FD(1) = ( X(1)**2 - X(2)**2 - X(3)**2
     +           + 2 * X(1) * ( X(2) + X(3) ) )
     +           / ( X(1) + X(2) + X(3) )**2
C
C ------ Partial derivative d(F) / d(X2)
         FD(2) = ( X(2)**2 - X(1)**2 - X(3)**2
     +           + 2 * X(2) * ( X(1) + X(3) ) )
     +           / ( X(1) + X(2) + X(3) )**2
C
C ------ Partial derivative d(F) / d(X3)
         FD(3) = ( X(3)**2 - X(1)**2 - X(2)**2
     +           + 2 * X(3) * ( X(1) + X(2) ) )
```

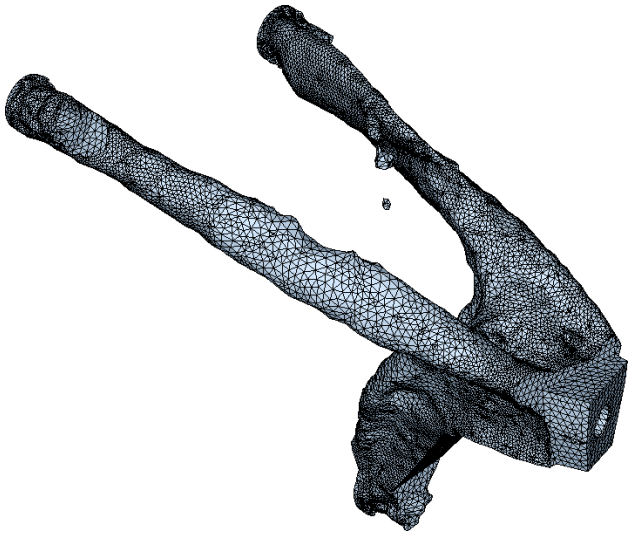
```
     +           / ( X(1) + X(2) + X(3) )**2
C
C ------ Weighting factors w_i
         W(1) = X(1) / ( X(1) + X(2) + X(3) )
         W(2) = X(2) / ( X(1) + X(2) + X(3) )
         W(3) = X(3) / ( X(1) + X(2) + X(3) )
C
C ****************
C INPUT FINISHED
C ****************
C
C
C ------ Writing output variables.txt (Schreiben des Ausgabefiles)
         I = I + 1
         IF (I .EQ. 1) THEN   ! 1: f,  2: df/dx
            WRITE(IUNIT,1100)W(1),W(2),W(3),X(1),X(2),X(3),
     +                       F,FD(1),FD(2),FD(3)
            WRITE(*     ,1100)W(1),W(3),W(3),X(1),X(2),X(3),
     +                       F,FD(1),FD(2),FD(3)
         ELSEIF (I .EQ. 2) THEN
            I = 0
         ENDIF
 1100 FORMAT(10(1X,E12.5))
C
      ELSE
         F = 0.0
         FD(1) = 0.0
         FD(2) = 0.0
         FD(3) = 0.0
         PRINT*, '***Function call not defined!'
      ENDIF
C
      RETURN
      END
```

C.1.4 Topology optimization with eigenfrequency constraint

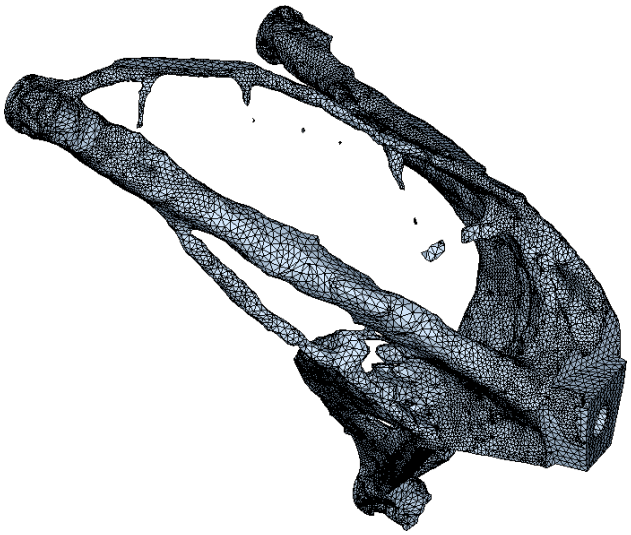
The effect of an eigenfrequency constraint in addition to a weight constraint as demonstrated in Fig. C.6. The convergence of the filling ratios is worsened and not realizable structures are created. An application of an eigenfrequency constraint as shown here is certainly not suitable.

C.1.5 Comparison of topology optimization and CAD model

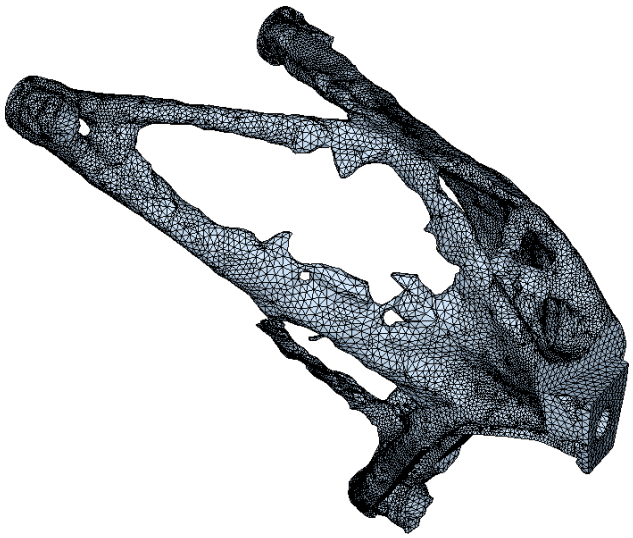
The comparison of the topology optimization variant *m80* and the derived CAD model as performed by the design engineer at Porsche is shown in more detail in Fig. C.7.



(a) *No eigenfrequency constraint*

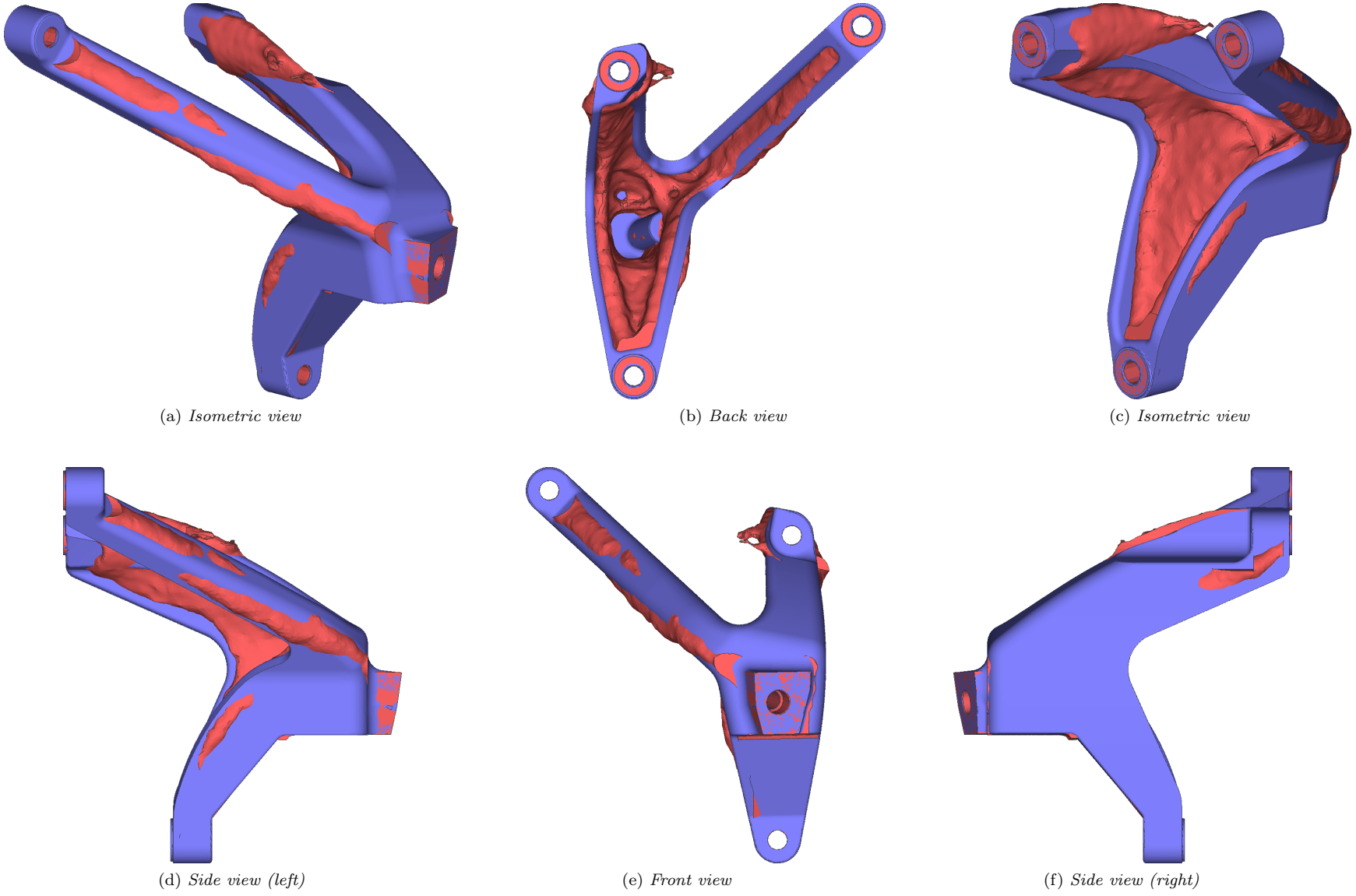


(b) *Eigenfrequency constraint* $\omega_1 \geq 500\,\mathrm{Hz}$



(c) *Eigenfrequency constraint* $\omega_1 \geq 800\,\mathrm{Hz}$

Figure C.6: *Topology optimization for an optimal stiff design with a mass constraint* $m \leq 0.8\,m_0$ *and with or without eigenfrequency constraints*



(a) *Isometric view* (b) *Back view* (c) *Isometric view*

(d) *Side view (left)* (e) *Front view* (f) *Side view (right)*

Figure C.7: *Comparison of the topology optimization variant m80 ($m = 0.8\,m_0$) and the CAD model ($m = 0.77\,m_0$)*

C.2 Multi-axial fatigue analysis with FEMFAT

The general data input and output of FEMFAT for a multi-axial fatigue analysis is shown in Fig C.8.

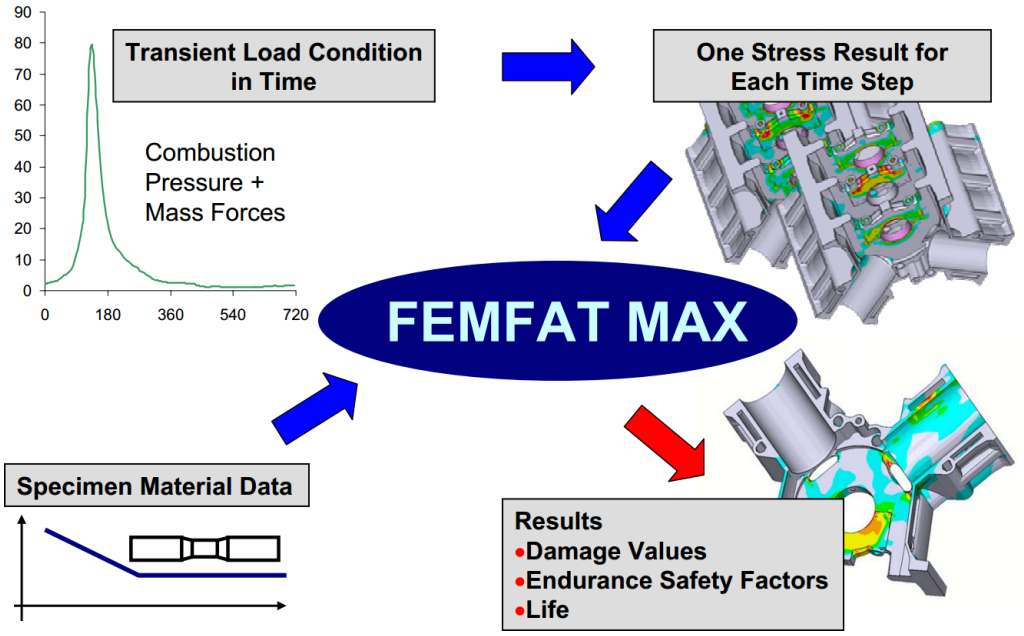
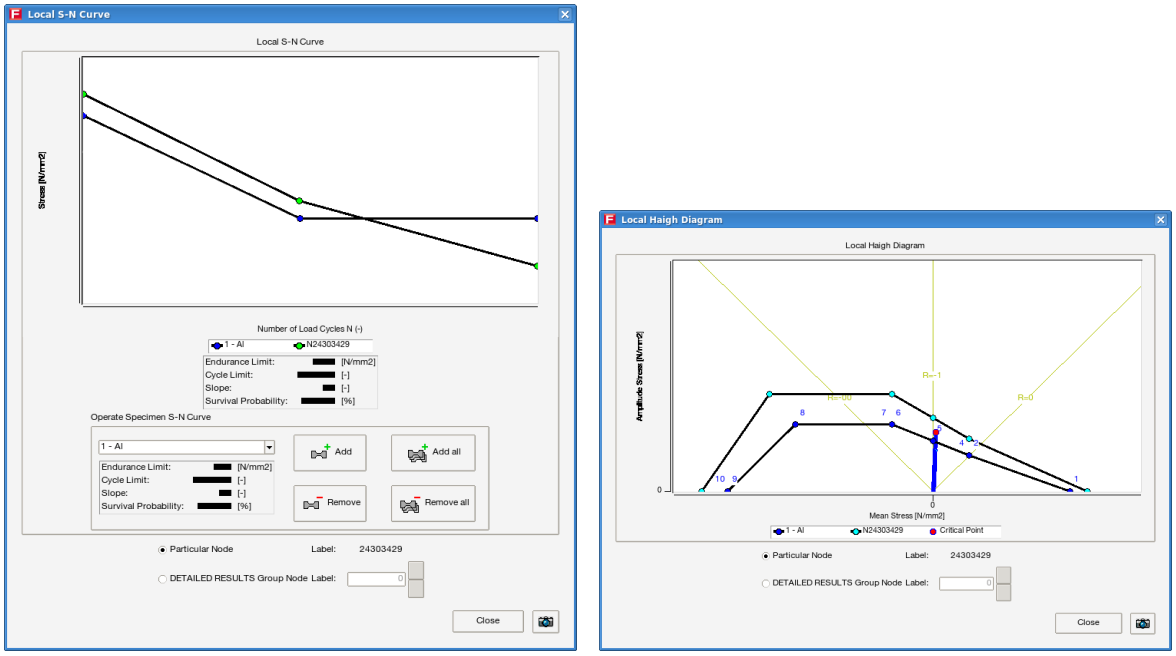


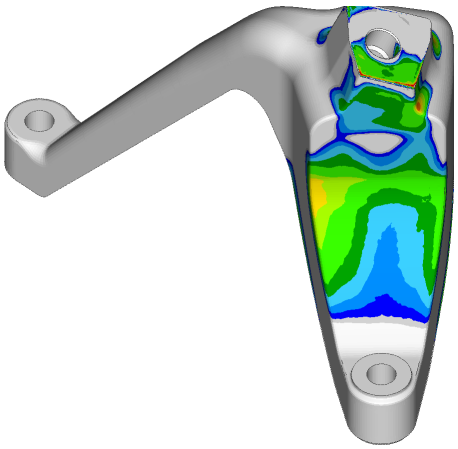
Figure C.8: *Exemplary input and output process of FEMFAT [28]*

An exemplary modified S-N curve and a modified Haigh diagram are depicted below. These curves are based on material measurement data and are computed for every FE node in the FEMFAT analysis.

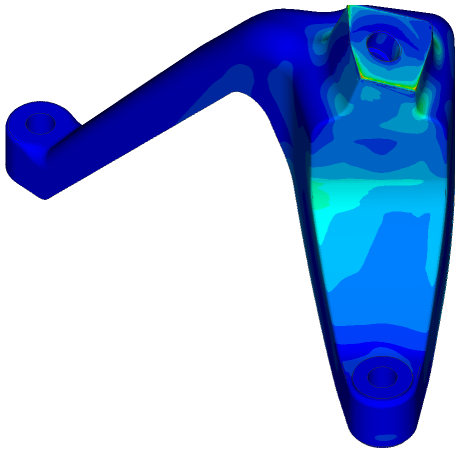


(a) *Local S-N curve* (b) *Local Haigh diagram*

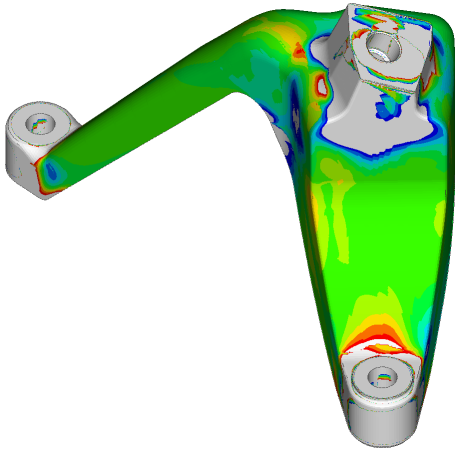
Figure C.9: *Modified S-N curve and modified Haigh diagram in FEMFAT*



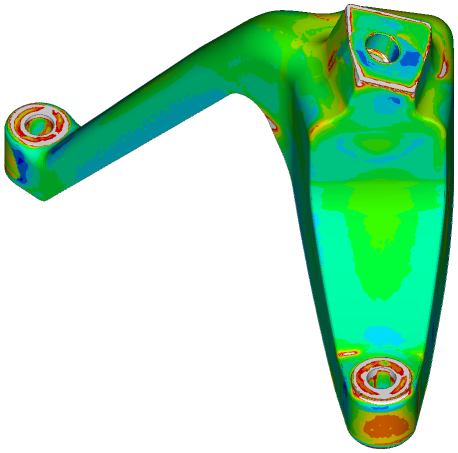
(a) *Safety factor* SF_A



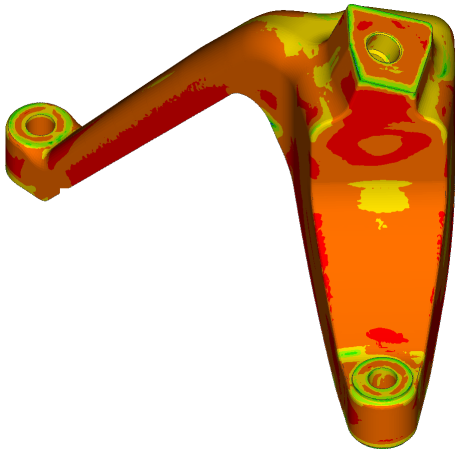
(b) *Amplitude stress* σ_a



(c) *Mean stress* σ_m



(d) *Local fatigue limit* FL



(e) *Influence of stress gradient on fatigue strength*

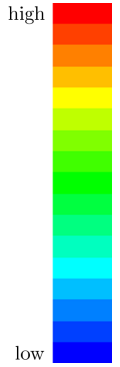


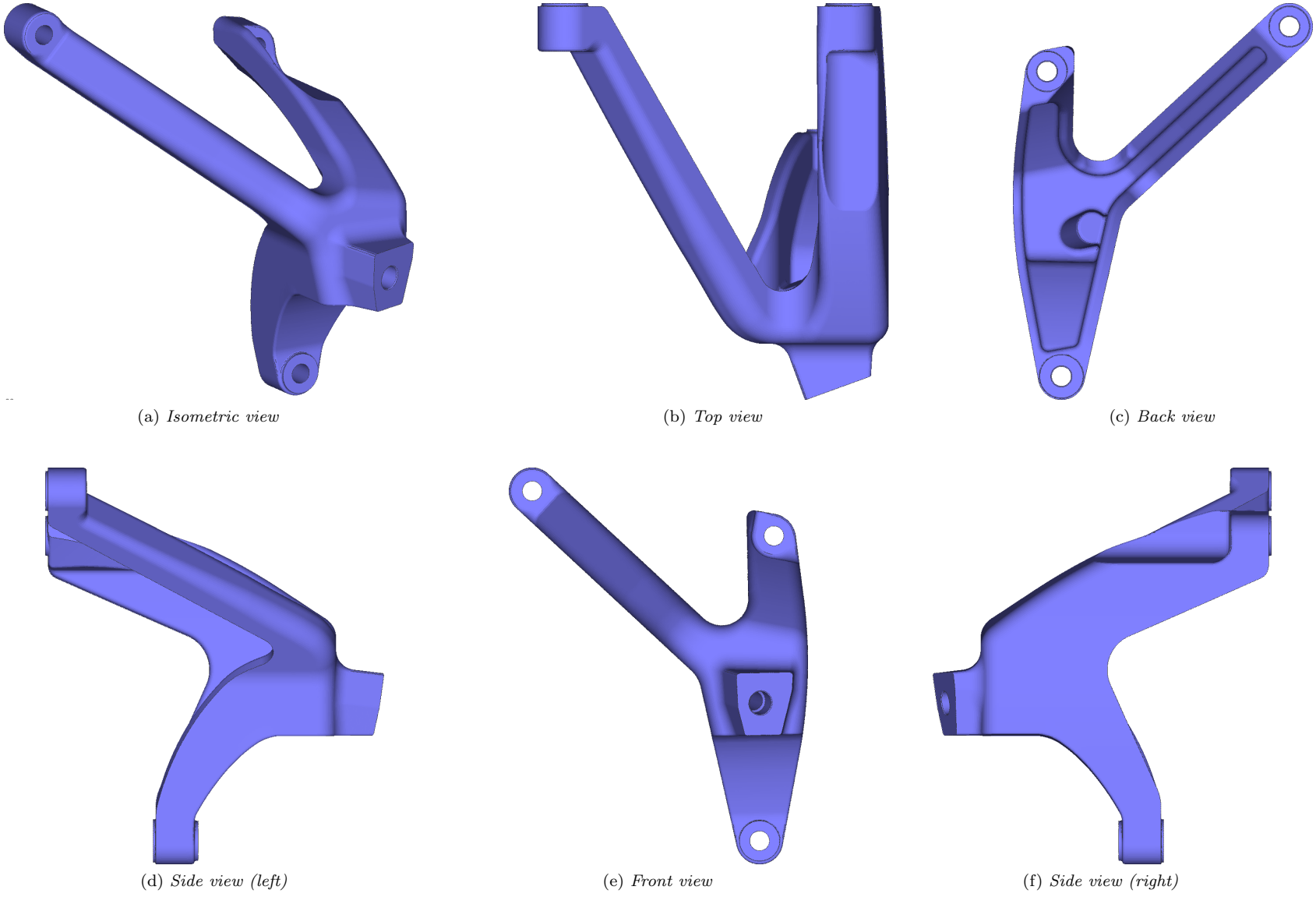
Figure C.10: *Additional FEMFAT results*

C.3 Final engine mount bracket design

To conclude this work, the final design of the new, weight-optimized engine mount bracket is illustrated from different perspectives in Fig. C.11.

The geometry is based on the topology optimization variant *m80* with $m = 0.8m_0$ that is transformed into a CAD model. With regards to operational strength that is assessed in a multi-axial fatigue analysis, a shape optimization is performed at the most critical spot which leads to a slight modification of a transition fillet where stress can be reduced. Furthermore, modal and linear buckling analyses are performed and approve the final design.

In total, the weight is reduced by 23 % in comparison to the old engine mount bracket design without a reduction of stiffness.



(a) *Isometric view* (b) *Top view* (c) *Back view*

(d) *Side view (left)* (e) *Front view* (f) *Side view (right)*

Figure C.11: *Final engine mount bracket design with* $m = 0.77\, m_0$

Index

References

[1] U. Kirsch. *Structural optimization: fundamentals and applications*. Springer-Verlag, 1993.

[2] P. Christensen and A. Klarbring. *An Introduction to Structural Optimization*. Springer, 2008.

[3] M. Bendsøe and O. Sigmund. *Topology Optimization: Theory, Methods and Applications*. Springer, 2003.

[4] J. Herskovits. *Advances in Structural Optimization*. Solid mechanics and its applications. Kluwer Academic Publishers, 1995.

[5] A. Belegundu and T. Chandrupatla. *Optimization Concepts and Applications in Engineering*. Cambridge University Press, 2011.

[6] L. Harzheim. *Strukturoptimierung: Grundlagen und Anwendungen*. Verlag Harri Deutsch, 2008.

[7] J. Arora. *Optimization of Structural and Mechanical Systems*. World Scientific, 2007.

[8] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[9] L. Råde and B. Westergren. *Mathematics Handbook*. Studentlitteratur, 2004.

[10] O. Zienkiewicz and J. Campbell. *Shape optimization and Sequential Linear Programming - Optimum Structural Design*. John Wiley & Sons, Ltd., 1973.

[11] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2006.

[12] C. Fleury and V. Braibant. "Structural optimization: A new dual method using mixed variables". In: *International Journal for Numerical Methods in Engineering* 23.3 (1986), pp. 409–428.

[13] C. Fleury. "CONLIN: An efficient dual optimizer based on convex approximation concepts". In: *Structural optimization* 1.2 (1989), pp. 81–89.

[14] K. Svanberg. "The method of moving asymptotes—a new method for structural optimization". In: *International Journal for Numerical Methods in Engineering* 24.2 (1987), pp. 359–373.

[15] C. Fleury. "Efficient approximation concepts using second order information". In: *International Journal for Numerical Methods in Engineering* 28.9 (1989), pp. 2041–2058.

[16] C. Fleury. "Sequential convex programming for structural optimization problems". In: *Optimization of Large Structural Systems*. Springer, 1993, pp. 531–553.

[17] H. Chickermane and H. Gea. "Structural optimization using a new local approximation method". In: *International Journal for Numerical Methods in Engineering* 39.5 (1996), pp. 829–846.

[18] M. Bendsøe. "Optimal shape design as a material distribution problem". In: *Structural optimization* 1.4 (1989), pp. 193–202.

[19] *PERMAS Example Manual*. Version 14.00.515. INTES GmbH. 2012.

[20] J. Xie and G. Steven. *Evolutionary Structural Optimization*. Springer, 1997.

[21] M. Zhou et al. "An integrated approach to topology, sizing and shape optimization". In: *Structural and Multidisciplinary Optimization* 26.5 (2004), pp. 308–317.

[22] M. Zhou, Y. Shyy, and H. Thomas. "Checkerboard and minimum member size control in topology optimization". In: *Structural and Multidisciplinary Optimization* 21.2 (2001), pp. 152–158.

[23] K.-J. Bathe and P. Zimmermann. *Finite-Elemente-Methoden*. Springer, 2002.

[24] *PERMAS User's Reference Manual*. Version 14.00.121. INTES GmbH. 2012.

[25] *Abaqus 6.13 Online Documentation*. Dassault Systèmes Simulia Corp. 2013.

[26] J. Reddy. *An Introduction to Continuum Mechanics*. Cambridge University Press, 2008.

[27] *FEMFAT 5.0 Documentation*. Magna Powertrain. 2012.

[28] *Workshop for Newcomer*. Magna Steyr AG & Co KG. 2005. URL: http://www.femfat.com/fileadmin/downloads/szdwnld.php?i=2305.

[29] *Bosch Automotive Handbook*. Bentley Publishers, 2007.

[30] C. Zillober. *SCP - an implementation of a sequential convex programming algorithm for nonlinear programming*. Schwerpunktprogramm anwendungsbezogene Optimierung und Steuerung: Report No. 470. Institut für Mathematik, Universität Bayreuth, 1993.

[31] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, Ltd., 2013.

[32] T. Chung and C. Chiou. "Structural shape optimization using self-adjusted convex approximation". In: *Structural and Multidisciplinary Optimization* 24.3 (2002), pp. 218–224.

[33] R. Horst, P. Pardalos, and H. Romeijn. *Handbook of Global Optimization*. Vol. 2. Handbook of Global Optimization. Springer, 2002.

[34] K. Lange. *Optimization*. Springer Texts in Statistics. Springer, 2013.

[35] W. Spillers and K. MacBain. *Structural Optimization*. Springer, 2009.

[36] H. Wang, Q. Ni, and H. Liu. "A new method of moving asymptotes for large-scale linearly equality-constrained minimization". In: *Acta Mathematicae Applicatae Sinica, English Series* 27.2 (2011), pp. 317–328.

[37] K. Schnittkowski and C. Zillober. "Sequential Convex Programming Methods". In: *Stochastic Programming*. Vol. 423. Springer, 1995, pp. 123–141.

[38] C. Zillober. "Global Convergence of a Nonlinear Programming Method Using Convex Approximations". In: *Numerical Algorithms* 27.3 (2001), pp. 265–289.

[39] A. Schumacher. *Optimierung mechanischer Strukturen: Grundlagen und industrielle Anwendungen*. Springer Vieweg, 2013.

[40] M. Bucalem and K. Bathe. *The Mechanics of Solids and Structures - Hierarchical Modeling and the Finite Element Solution*. Computational Fluid and Solid Mechanics. Springer, 2011.