Structural optimization of components and joints in assemblies considering fail-safety

Vom Promotionsausschuss der
Technischen Universität Hamburg

zur Erlangung des akademischem Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

von
Olaf Bernhard Ambrozkiewicz

aus
Hamburg

2022

Erster Gutachter: Prof. Dr.-Ing. Benedikt Kriegesmann
Zweiter Gutachter: Prof. Dr.-Ing. Robert Seifried

Tag der mündlichen Prüfung: 30. September 2022

DOI: 10.15480/882.4618
ORCID iD: 0000-0003-3250-7095

To my family

Abstract

Structural optimization has become an increasingly important part of product development, especially in the aerospace industry, where weight savings due to lightweight design have a particularly strong impact on efficiency and thus economy and environmental compatibility. One area of structural optimization is topology optimization, which offers maximum design freedom and thus enables the greatest improvements. However, load-adapted designs obtained by topology optimization are usually highly sensitive to an unpredictable local loss of stiffness, like e.g. for the case of randomly inflicted damage to individual load paths of the structure. Therefore, these designs are not considered fail-safe.

This thesis presents a two-stage procedure for density-based optimization towards a fail-safe design. Existing approaches are either computationally extremely expensive or do not explicitly consider fail-safe requirements in the optimization. The presented method trades off both aspects by employing a two-stage optimization approach to provide redundant designs that offer robustness to the failure of single load paths. In the first stage, a topology optimization with local volume constraints is performed. The second stage is referred to as "density-based shape optimization" since it only alters the outline of the structure while still acting on a fixed voxel-type finite element mesh with pseudo-densities assigned to each element. The performance gain and computational efficiency of the proposed method are demonstrated by application to various 2D and 3D examples. The results show, that the presented method can be carried out with reasonable computational effort, in contrast to existing approaches with explicit consideration of fail-safety in topology optimization. For the 2D examples considered, the number of analyses for a fail-safe optimization is reduced by three orders of magnitude compared to existing methods and is at most 5.6 times higher than for a standard topology optimization. Consequently, the proposed method is also applicable for large-scale models in an industrial context.

With the possibility to compute and manufacture single optimized components, the question of how to optimize the connections between different components in an assembly arises. This thesis therefore also provides a method for the simultaneous optimization of the topology of components and their corresponding joint locations in an assembly. Therein, the joint locations are not discrete and predefined, but continuously movable. The underlying coupling equations allow for connecting dissimilar meshes and avoid the need for remeshing when joint locations change. The presented method models the force transfer at a joint location not only by using single spring elements but accounts for the size and type of the joints. When considering e.g. riveted or bolted joints, the local part geometry at the joint location consists of matching holes that are surrounded by material. For spot welds, the joint locations are filled with material and may be smaller than for bolts. The presented method incorporates these material and clearance zones into the simultaneously running topology optimization of the components. Furthermore, failure of joints may be taken into account at the optimization stage, yielding assemblies connected in a fail-safe manner. Finally, by embedding the above-mentioned efficient method for fail-safe optimization of single components in the presented assembly optimization framework, damage tolerant assemblies can be obtained that are robust to the failure of joints and single load paths of each component.

Keywords: fail-safety; joint optimization; topology optimization; shape optimization

Kurzfassung

Die Strukturoptimierung ist ein zunehmend wichtiger Bestandteil der Produktentwicklung geworden, vor allem in der Luft- und Raumfahrt, wo sich Gewichtseinsparungen durch Leichtbau besonders stark auf die Effizienz und damit auf die Wirtschaftlichkeit und Umweltverträglichkeit auswirken. Ein Gebiet der Strukturoptimierung ist die Topologieoptimierung, die maximale Gestaltungsfreiheit bietet und damit die größten Verbesserungen ermöglicht. Lastangepasste Entwürfe aus einer Topologieoptimierung sind allerdings in der Regel sehr empfindlich gegenüber einem unvorhersehbaren lokalen Steifigkeitsverlust, etwa durch zufällig auftretende Schäden an einzelnen Lastpfaden der Struktur. Diese Entwürfe gelten daher nicht als ausfallsicher.

In dieser Arbeit wird ein zweistufiges Verfahren zur dichtebasierten Optimierung von ausfallsicheren Bauteilen vorgestellt. Bestehende Methoden sind entweder extrem rechenintensiv oder berücksichtigen die Anforderungen an die Ausfallsicherheit nicht explizit in der Optimierung. Der vorgestellte Ansatz stellt einen Kompromiss zwischen beiden Vorgehensweisen dar, indem er mit Hilfe einer zweistufigen Optimierung redundante Entwürfe liefert, die robust gegenüber dem Ausfall einzelner Lastpfade sind. In der ersten Stufe wird eine Topologieoptimierung mit lokalen Volumennebenbedingungen durchgeführt. Die zweite Stufe wird als „dichtebasierte Formoptimierung“ bezeichnet, da sie nur die Umrisse der Struktur verändert, die weiterhin durch Pseudodichten in einem voxelartigen Finite-Elemente-Netz beschrieben ist. Die erzielbaren Verbesserungen und der damit verbundene Rechenaufwand durch Anwendung der vorgeschlagenen Methode werden anhand verschiedener 2D- und 3D-Beispiele demonstriert. Die Ergebnisse zeigen, dass die vorgestellte Methode im Gegensatz zu bestehenden Methoden mit expliziter Berücksichtigung von Ausfallsicherheit in der Topologieoptimierung mit vertretbarem Rechenaufwand durchgeführt werden kann. Für die gezeigten 2D-Beispiele verringert sich so etwa die Gesamtzahl der Rechnungen für eine Optimierung hinsichtlich Ausfallsicherheit um drei Größenordnungen und ist maximal 5,6-mal höher als bei einer gewöhnlichen Topologieoptimierung. Das unterstreicht, dass die vorgestellte Methode auch für große Modelle im industriellen Kontext anwendbar ist.

Mit der Möglichkeit einzelne optimierte Komponenten zu berechnen und herzustellen ergibt sich auch die Frage, wie die Verbindungen zwischen verschiedenen Komponenten in einer Baugruppe optimiert werden können. In dieser Arbeit wird daher zudem auch eine Methode zur gleichzeitigen Optimierung der Topologie von Bauteilen und deren Verbindungen in Baugruppen vorgestellt. Dabei ist die Lage der Verbindungsstellen weder diskret noch vorgeschrieben, sondern kontinuierlich beweglich. Kopplungsgleichungen ermöglichen den Einsatz unterschiedlich vernetzter Entwurfsräume und umgehen eine Neuvernetzung, wenn sich die Positionen der Verbinder ändern. In der vorgestellten Methode ist die Kraftübertragung an einer Verbindungsstelle nicht nur durch einzelne Federelemente modelliert, sondern es werden auch die Größe und Art der Verbindung berücksichtigt. Für Niet- oder Schraubverbindungen müssen beispielsweise lokal an den Bauteilen entsprechende passgenaue Löcher, umgeben von Material, vorhanden sein. Punktschweißungen hingegen können kleiner ausfallen und erfordern nur das Vorhandensein von Material. Die vorgestellte Methode bezieht diese Material- und Freibereiche in die gleichzeitig laufende Topologieoptimierung der Bauteile mit ein. Darüber hinaus kann das Versagen von Verbindern schon während der Optimierung berücksichtigt werden, so dass sich ausfallsicher verbundene Baugruppen ergeben. Durch die Einbettung der effizienten Methode zur ausfallsicheren Optimierung von Einzelkomponenten in die vorgestellte Baugruppenoptimierung können schließlich schadenstolerante Baugruppen errechnet werden, die robust gegenüber einem Ausfall von Verbindern sowie von Lastpfaden der einzelnen Komponenten sind.

Schlagworte: Ausfallsicherheit; Verbindungsoptimierung; Topologieoptimierung; Formoptimierung

Contents

Abbreviations

2D	Two-dimensional
3D	Three-dimensional
AL	Augmented Lagrange
AM	Additive manufacturing
BESO	Bidirectional Evolutionary Structural Optimization
CAD	Computer Aided Design
CDF	Cumulative distribution function
DOF	Degree(s) of freedom
DS	Design space
DV	Design variable
ESO	Evolutionary Structural Optimization
FCRC	Flight Crew Rest Compartment
FE	Finite element
FEM	Finite element method
FS	Fail-safe
GL1, GL2, GLn	Grid-level 1, ..., n
GVC	Global volume constraint
HS	Hashin-Shtrikman
KS	Kreisselmeier-Steinhauser
LVC	Local volume constraint(s)
MMA	Method of Moving Asymptotes
NDS	Non-design space
PDF	Probability density function
RAMP	Rational Approximation of Material Properties
RDM	Rectangular damage model
RDO	Robust design optimization
RTO	Robust topology optimization
SIMP	Solid Isotropic Material with Penalization
SKO	Soft Kill Option (algorithm)
SO	Shape optimization
TO	Topology optimization

Symbols used

$\boldsymbol{a}, a, b, c$	Lengths of semi-axes of a shape as vector or in components
a_0, a_i	Coefficients used in the MMA
A	Generic, smooth aggregation function
A_g	General, "p-like" aggregation function
A_{KS}	KS aggregation function
A_{pm}	P-mean aggregation function
A_{pn}	P-norm aggregation function
A_{uni}	Aggregation function for uniformly distributed values
b	Width of a solid feature
c	Compliance objective function
c_0	Nominal (undamaged) compliance
c_c	Compliance of connections
c_c^i	Compliance of joint i
c_d	Compliance for damage scenario d
c_d^m	Compliance for damage scenario d and failure mode m
c_i	Coefficients used in the MMA
c_m	Compliance of material
c_m^i	Compliance of part i's material
$\boldsymbol{c}^i$	Location vector of element i's center point
$\boldsymbol{C}$	Matrix with center points
$\bar{c}$	Maximum (worst-case) compliance value
$\bar{c}_J, \bar{c}_{LP}$	Maximum compliance value for a failure of joints or load paths
$\bar{c}^1$, $\bar{c}^2$	Maximum compliance values for failure mode $m = 1$ or $m = 2$
$\bar{c}^1$, $\bar{c}^2$, $\bar{c}^3$, $\bar{c}^f$	Maximum compliance values for damage sets GL1, GL2, GL3, and the full set
$\bar{c}_t$	Threshold value for maximum compliance
d_0	Minimum allowed distance
d_i	Coefficients used in the MMA
d^{ij}	Absolute distance between reference points i and j
E	Superellipsoid function
E_0	Young's modulus of solid phase
E_i	Interpolated Young's modulus of element i
E_{min}	Young's modulus of void phase
f	Probability density function
F	Cumulative distribution function
$\boldsymbol{f}$	External force vector
$\tilde{\boldsymbol{f}}$	Augmented external force vector
$\boldsymbol{f}_{inner}^i$	Inner nodal forces of joint i
$\boldsymbol{F}^i, F_x^i, F_y^i, F_z^i$	Resultant inner force on one part for joint i and its components
$\bar{F}_t$	Threshold value for maximum joint force
$\boldsymbol{f}_m$	External forces on the parts
$\boldsymbol{f}_m^i$	External forces on part i
g_d, g_c, g_s	Discrete, piece-wise constant, and smooth progressive mapping functions
$\boldsymbol{g}^i$	Coupling term matrix for joint i
$\boldsymbol{G}$	Coupling term matrix
h, h_i, h_j	Inequality constraint functions
h_{agg}	Aggregated inequality constraint function

i, j, k	Indices
k_c	Resultant joint stiffness
$\boldsymbol{k}_{c,0}$	Vector of all resultant joint stiffnesses for intact joints
$\boldsymbol{k}_{c,i}$	Vector of all resultant joint stiffnesses for joint failure case i
k_g	Global volume fraction upper bound
k_l	Local volume fraction(s) upper bound(s)
k_p	Global volume fraction upper bound, part-wise
$\boldsymbol{k}_0$	Unitary stiffness matrix of an element with $E = 1$
$\boldsymbol{k}_i$	Stiffness matrix of element i
$\boldsymbol{K}$	Global stiffness matrix of system
$\tilde{\boldsymbol{K}}$	Augmented global stiffness matrix of system
$\boldsymbol{K}_c$	Connection stiffness part of augmented global system matrix
$\boldsymbol{K}_c^i$	Connection stiffness matrix of joint i
$\boldsymbol{K}_i$	Global stiffness matrix for a configuration i
$\boldsymbol{K}_m$	Material part of augmented global system matrix
$\boldsymbol{K}_m^i$	Material stiffness matrix of part i
$\boldsymbol{K}^i$	Stiffness matrix of part i
$\boldsymbol{l}_x^i, \boldsymbol{l}_y^i, \boldsymbol{l}_z^i$	Selection vectors for x, y, or z terms for joint i
m	Failure mode
$\bar{m}$	Maximum considered failure mode
n	Index variable
$\tilde{n}$	Normalized index variable
n_c	Number of constraints
n_d	Number of damage scenarios
n_{dJ}	Number of joint damage scenarios
n_{dL}	Number of load path damage scenarios
n_{DOF}	Number of degrees of freedom
n_e	Number of elements
n_j	Number of joints
n_n	Number of nodes for a single element
n_p	Number of parts
n_t	Number of terms
N_i	i-th element shape function
p	SIMP material penalization parameter
$p, \hat{p}$	Aggregation function parameter
$\boldsymbol{p}$	Combined pseudo load vector for adjoint system regarding joint forces
$\boldsymbol{p}^i$	Pseudo load vector for adjoint system regarding forces of joint i
q	RAMP material penalization parameter
q	Input distribution modeling function parameter
q	Shape exponent of a superellipsoid
r	Filter radius
r'	NDS zone radius
R	Local volume constraint(s) neighborhood radius
s	Positive scaling factor
s^{ij}	Squared distance between reference points i and j
S	Sum term
t	Generic threshold value
$\boldsymbol{u}$	Global displacement vector

$\tilde{\boldsymbol{u}}$	Augmented global displacement vector
$\boldsymbol{u}_c$	Nodal displacements of all joints
$\boldsymbol{u}_c^i$	Nodal displacements of joint i
$\boldsymbol{u}_m$	Nodal displacements of all parts
$\boldsymbol{u}_m^i$	Nodal displacements of part i
$\boldsymbol{u}_e, \boldsymbol{u}, \boldsymbol{u}_d$	Displacement vector for the eroded, nominal, and dilated design
v_i	Volume of element i
V	Global volume of the part
V_0	Volume of the design space
$\boldsymbol{V}, \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3$	Principal axes of a shape as orthogonal matrix or normalized vectors
w	Filter weighting function
w_i	Weight of term i in a linear combination of terms
$\boldsymbol{x}$	Design variables for joints
$\boldsymbol{x}'$	Position vector in local coordinates of a shape
$\boldsymbol{x}_c$	Center point location of a shape
x_l, x_u	Lower and upper bound values for x coordinates
$\boldsymbol{x}^i, x^i, y^i$	Location vector of joint i and its components
$\boldsymbol{y}$	Auxiliary variable vector used in the MMA
y_l, y_u	Lower and upper bound values for y coordinates
z	Auxiliary variable used in the MMA
$\mathbb{L}_i$	Set of element indices for part i
$\mathbb{N}_i$	Set of element indices in a neighborhood around element i
$\mathbb{S}$	Set of element indices for an identified cluster
α	Correction factor or prefactor
α	Shape projection steepness parameter
β	Density projection steepness parameter
ϵ	Small value
$\eta, \boldsymbol{\eta}$	Projection threshold parameters, scalar or element-wise
η_e, η_0, η_d	Projection threshold values for the eroded, nominal, and dilated design
κ	Material stiffness interpolation function
$\boldsymbol{\lambda}$	Adjoint vector for sensitivity calculation
$\boldsymbol{\Lambda}, \lambda_1, \lambda_2, \lambda_3$	Eigenvalue matrix and eigenvalues
$\boldsymbol{\lambda}_c$	Lagrange multipliers for couplings
$\boldsymbol{\lambda}_c^i$	Lagrange multipliers for couplings of joint i
ν	Poisson's ratio
$\boldsymbol{\xi}$	Element's local coordinate vector
$\boldsymbol{\varrho}$	Design variables for TO
$\boldsymbol{\varrho}^i$	Design variables for TO for part i
$\tilde{\boldsymbol{\varrho}}$	Filtered variables for TO
$\bar{\boldsymbol{\varrho}}$	Projected variables for TO
$\bar{\boldsymbol{\varrho}}_{comp}$	Complementary field of projected variables
$\bar{\boldsymbol{\varrho}}_e, \bar{\boldsymbol{\varrho}}, \bar{\boldsymbol{\varrho}}_d$	Projected variables for the eroded, nominal, and dilated design
$\bar{\varrho}_t$	Threshold value for projected variables
$\hat{\boldsymbol{\varrho}}$	Modified variables for TO
$\hat{\boldsymbol{\varrho}}_{comp}$	Complementary field of modified variables
$\hat{\boldsymbol{\varrho}}_d$	Modified variables for damaged configuration d
σ	Standard deviation
$\boldsymbol{\sigma}$	Stress tensor

$\overline{\sigma}, \underline{\sigma}$	Highest and second highest principal stress magnitude
$\boldsymbol{\Sigma}$	Covariance matrix of center points' coordinates
τ	Modeling function for input terms
τ_{agg}	Aggregated term
$\boldsymbol{\tau}, \boldsymbol{\mathcal{T}}$	Vector of terms (dependent on the DVs) to be aggregated
$\bar{\tau}$	Maximum value of all considered terms
φ_n	Normalized stress criterion
Φ	Objective function
Φ_{agg}	Aggregated objective function
Φ_i	Objective function for a scenario i
Φ_d	Objective function for damaged configuration d
Φ_d^m	Objective function for damaged configuration d and failure mode m
$\boldsymbol{\psi}$	Element-wise defined mask
$\boldsymbol{\psi}_{comp}$	Complementary mask
$\boldsymbol{\psi}^i$	Mask for joint i
$\boldsymbol{\psi}_J$	All masks for all joints
${}^{-}\boldsymbol{\psi}$	Mask for material removal
${}^{+}\boldsymbol{\psi}$	Mask for material introduction
${}^{-}\boldsymbol{\psi}_d$	Damage mask for damage case d
ω	Global volume fraction
ω_i	Local volume fraction within a neighborhood $\mathbb{N}_i$
$\boldsymbol{\omega}$	Vector of all local volume fractions
Ω	Design space for TO
${}^{-}\Omega_d$	Damaged region for failure scenario d

1 Introduction

Lightweight design of structures becomes increasingly important as it enables more efficient use of finite resources. If a part can be manufactured with less material, its weight is also reduced. Especially in the context of mobility, weight reduction means savings in fuel and thus in operational expenses, as it is prominently the case for aircrafts [1].

As a tool to generate lightweight, load-adapted designs that outperform manually designed parts, topology optimization has made its way from academia to industrial application. Often, topology optimized designs are described as having a "bionic" or "organic" shape.

However, these designs usually differ from bio-inspired or biomimetic designs. Topology optimized designs are calculated by mathematical means in defining the optimization task as a minimization of an often fairly simple objective function. In contrast to that, structures in nature are far more complex and need to fulfill a wide variety of different tasks besides just having e.g. an optimized mechanical behavior. For the example of bone microstructure, this is discussed in detail in [2]. Consequently, designs obtained from a topology optimization are considered as being closer to optimal when compared to biomimetic designs when considering mechanical properties only.

Designs derived from a topology optimization that were once considered as too difficult to manufacture using conventional manufacturing processes are now becoming reality as additive manufacturing (AM) is recognized as a mature manufacturing process [3]. Consequently, motivated by the advances in AM, design optimization and in particular topology optimization is increasingly used for the design of lightweight structures, especially in the aerospace industry (see e.g. [4, 5]). One of the most prominent examples is the Flight Crew Rest Compartment (FCRC) bracket of the Airbus A350, which has been replaced by an additively manufactured part [6, 7], as shown in figure 1.1.

(a) Conventional part **(b)** Additively manufactured part

Figure 1.1 – FCRC bracket, conventional and lightweight part, from [7].

The concurrent advances in topology optimization methods on the one hand and manufacturing processes on the other hand facilitate an ever-growing field of application for topology optimization [8]. As a result, possible applications of topology optimization reported in the literature cover use cases beyond classical mechanical engineering, ranging from nano-scale structures used in photonics (see e.g. [9]) up to large-scale structures in civil engineering (see e.g. [10]). Nevertheless, even in the well-studied purely mechanical context of compliance minimization problems, which is also the focus of this work, unresolved issues and knowledge gaps remain.

This work goes beyond the state of the art in presenting methods to augment the basic workflow of a density-based topology optimization by two main aspects, each of which is highly relevant for practical applications: First, it addresses the problem that in safety-critical

applications a structure is required to be fail-safe. This means that it must still offer a certain mechanical performance even in case of partial failure due to damage. Second, a method for simultaneously optimizing multiple parts and their interconnections that form an assembly is presented. Both aspects are further discussed in the following section 1.1.

The focus of this work is to develop methods that tackle the above two problems in a feasible amount of computational time such that practical application is not compromised.

1.1 Motivation

In viewing a plain topology optimization as a tool to remove inefficient material, it is easy to imagine that the gain in structural performance versus the weight of the optimized part often comes at the price of reduced robustness with relation to a variation of modeling parameters. For instance, the stiffness of a part being optimized for a certain load direction is negatively affected by randomly misaligned loads if this uncertainty of loading is not accounted for in the optimization [11]. Hence, for lightweight design, there exists a conflict of interest between maximal structural performance at the least weight and robustness with respect to unmodeled deviations in e.g. loads, geometry, or material properties.

An extreme case of such deviations is structural damage, in which parts of a structure lose their load-bearing capability or the latter is at least severely impaired. Especially for the aircraft industry, strict rules apply regarding the damage tolerance of parts [12]. One such requirement is that safety-critical structures need to be fail-safe. For instance, according to Niu [13], fail-safety means that designated loads are supported and sufficient residual stiffness remains in case of failure of a single member or partial damage. Additionally, the Certification Specifications for Large Aeroplanes CS-25 [14] requires a "multiple load path construction" for certain airframe structures, as well as the proof that "the aeroplane may function safely with an element missing". In other words, these structures need to be fail-safe by being able to resist a certain load level, even if one load path fails.

Therefore, it is desirable to develop methods for lightweight design that consider fail-safe requirements already at the optimization stage. However, the computational effort of existing methods is very high since a large number of analyses is required in each optimization iteration. Consequently, already small academic examples may demand an infeasible amount of computational time. To be able to perform fail-safe optimizations with high-resolution FE models encountered in an industrial context, computational savings of several orders of magnitude are needed to make these problems solvable in practice. This motivates the development of alternative, efficient methods for fail-safe optimization that are presented in this work.

Another important aspect for the use of topology optimized designs in practice is, that most engineering structures are not stand-alone parts but instead are composed of multiple parts, interconnected by e.g. fasteners or welds. Thus, topology optimizing only single parts in an assembly alters the load distribution at the interfaces and consequently also the loading in neighboring parts. Therefore, exchanging single "conventional" parts by topology optimized lightweight parts usually requires an updated loading simulation for adjacent, unchanged parts as well. The locations (or even sizes) of the connecting fasteners should also be reconsidered to adapt them to the changed loading situation. Consequently, to make use of the full potential of topology optimization of assemblies, the system boundary for the numerical simulations involved needs to be extended to also include neighboring parts and their connections.

Following this approach and optimizing assemblies as a whole yields different results compared to an approach that considers multiple single-part optimizations, one after the other, which can clearly only lead to suboptimal results. However, the topology optimization of assemblies

including their connections gained relatively little attention in the literature. Known methods for assembly optimization that worked for academic examples with very coarse FE mesh discretizations fail for the high-resolution models used nowadays. To develop a holistic approach for assembly optimization, the aspect of fail-safety at the assembly level should also be addressed in order to obtain connections between individual parts that are robust with respect to a failure of single connecting elements like e.g. bolts.

Considering the aforementioned aspects, topology optimization methods for assembly optimization are taken up in this work, adapted, and finally extended to a use for optimizations towards fail-safely connected assemblies.

1.2 State of the art

This work focuses on topology optimization methods that find an optimized material distribution layout of structures by solving a mathematical problem via gradient-based optimization algorithms[1]. Current implementations of gradient-based topology optimization methods are able to handle optimization problems involving billions of finite elements [10, 19–21].

Topology optimization problems have been approached by gradient-based algorithms first with homogenization methods [22] and later with a "power-law approach" [23] that incorporates material modeling via artificial material densities. This density-based approach is also adapted throughout this work. A comparative review of several alternative approaches for topology optimization not relevant for this work (like e.g. the level set approach [24, 25]) is given in [26].

For the density-based topology optimization method, problems arise concerning the non-existence of optimal solutions, numerical instabilities like checkerboard patterns, and mesh dependency, which is discussed in detail in [27]. To address these problems, at first, a gradient filter [28] that manipulated only the sensitivities used during optimization was proposed. This sensitivity filter was later expanded to a consistent density filter [29, 30]. In combination with a projection step introduced in [31], the widely used three field topology optimization approach is obtained that reflects the current state of the art for density-based topology optimization and is thus the basis for all algorithms presented in this work.

1.2.1 Topology optimization with consideration of partial damage

Embedding the fail-safe requirements in a structural optimization of a system composed of discrete members is straightforward and has already been done by Sun et al. in 1976 [32]. Therein, the aspect of fail-safety in size optimizations of truss structures was addressed by applying stress, buckling, and displacement constraints for each failure case. Due to the large number of constraints and computational limits, this approach is restricted to relatively small examples. For larger models, Sun et al. only considered a few failure cases at the same time.

Lüdecker and Kriegesmann [33] presented a method for the fail-safe size optimization of a beam lattice structure considering stress constraints and showed that this is a difficult task even for today's hardware, such that model size was also limited using their approach. Similar findings are also reported by Stolpe [34] for the less challenging problem of a compliance-based fail-safe truss lattice optimization.

[1] Alternative, non-gradient-based topology optimization methods not used in this work are among others: The Soft Kill Option (SKO) algorithm [15], which is based on a heuristic biological update rule, the Evolutionary Structural Optimization (ESO) method [16], which yields binary designs based on a heuristic criterion to remove inefficient material, or its bidirectional form BESO [17], also allowing material reintroduction. The latter method may however also be implemented to incorporate an update rule relying on sensitivity information [18].

While above-mentioned methods considered systems with an already established layout of discrete load-bearing members, implementing fail-safe requirements in a topology optimization raises the question of how to remove structural elements when these have yet to evolve during the optimization. Jansen et al. [35] tackled this problem by introducing a simplified damage model using damage patches of fixed rectangular shape, in which material is removed to model the effect of a partial failure. Since the location of the most critical damage is not known a priori, a vast number of possible damage locations has to be evaluated in each iteration. The result is a structure with redundant load paths, but the computational cost is extremely high. Considering that the number of possible damage locations is linked to the discretization of the design space for the topology optimization, the computational effort drastically increases for higher resolution FE meshes, especially for 3D problems.

A straightforward way to cut down the computational cost is to reduce the number of damage scenarios considered within each optimization iteration. Zhou and Fleury [36] presented an approach to down-select the locations at which damage shapes are to be placed in considering only a sparser distributed subset of possible damage locations. This reduced the computational cost that however remains on a fairly high level. Furthermore, their approach produces designs that are sensitive to damage at locations that have been sorted out, as first shown by the author of this thesis in [37].

To remedy the deficiency that critical damage locations may be skipped, most recently, Hederberg and Thore [38] combined the idea of the approach of Zhou and Fleury with an inner anti-optimization loop that refines the locations of a grid of sparsely distributed damage shapes in moving them to the most critical locations. However, this nested optimization comes again at a significantly increased number of FE calculations due to the inner optimization loop.

Another possibility to reduce the number of simulations is to combine failure cases in order to improve computational efficiency. However, Lüdeker and Kriegesmann [33] showed that this is also not necessarily a conservative approach, and moreover, the number of analyses to be run per iteration remains unacceptably high.

The author of this work presented an active-set strategy to adaptively disregard potentially uncritical damage locations in [37] for fail-safe topology optimization problems employing the rectangular damage model from Jansen et al. In this “threshold method”, all empty and not yet fully converged regions of the design space are excluded from damage considerations. While this method proved to be conservative, the number of analyses could only be reduced by about 40% for the examples considered. Wang et al. [39] followed the same idea and disregarded damage scenarios based on the stress state of the intact structure inside each damage shape and reported a similar reduction in computational cost. Thus, the achievable savings for both strategies are again not high enough.

Considering all aforementioned aspects, the author of this work moved away from the rectangular damage model from Jansen et al. and presented a method for fail-safe optimization where structural members are identified as “beams” and “knots” during the topology optimization [37]. Based on this identification, the damage patch size, shape, and location are adaptively adjusted within the optimization to cover all load-bearing parts of the structure, preferably in a minimal amount of damage patches. In the author’s opinion, this approach is close to the intention of the Certification Specifications for Large Aeroplanes [14]. The number of damage patches to be considered per iteration decreases by orders of magnitude when compared to the method in Jansen et al. However, since the optimization problem is continuously changed, the topology optimization does not converge reliably in all cases.

All above-mentioned methods have in common, that they tackle fail-safe optimizations in a “worst-case” way. This means, that the optimization problem is driven by the damage scenario

with the largest impact on the degradation of the load-bearing capability of the structure. This is a very conservative approach since the worst-case scenario is always considered, no matter how unlikely it is.

Alternatively, probabilistic approaches can be used, accounting for the likelihood of occurrence of each failure case, as presented by Cid et al. [40, 41]. While their reliability-based approach is a way to counter overly conservative, heavy designs, it however requires solid data on the probability of occurrence for every damage scenario to be available. An alternative, risk-averse approach is followed by Martinez-Frutos and Ortigosa [42], where risk functions, describing the cost inherent to partial failures, are incorporated into the objective function as a way to assign different weights to the individual failure scenarios. In a subsequent publication [43], the aspect of robustness is included by considering a weighted objective incorporating the mean value and standard deviation as a way to also control the scatter of the structural performance under partial failure. Nevertheless, when following probabilistic approaches, it depends on the chosen parameters if either a fail-safe design with multiple, redundant load paths or a design similar to one from a plain topology optimization without damage consideration is obtained.

Besides explicitly formulating the fail-safe requirement as objective or constraint in a topology optimization, a redundant structure can also be achieved by using the local volume constraints introduced by Wu et al. [44]. These constraints penalize large accumulations of solid volume in the design space and therefore enforce structural members to be distributed over the whole design space. While this approach is extremely efficient, the evaluation of failure scenarios is not part of the optimization and hence the fail-safe behavior of the resulting structure is not optimal and the desired damage tolerance cannot be guaranteed. However, the obtained structures involve redundant load paths and have therefore at least the potential to be more robust towards partial damage than a design obtained from a plain topology optimization, which is shown for different objective functions in [44] and [45].

1.2.2 Topology optimization of assemblies

The application of topology optimization in industry is in the majority of cases still limited to a replacement of single, conventionally designed parts inside their original assembly by topology optimized parts. For the aircraft industry, secondary structures like connecting brackets are in the focus of such measures to reduce weight [46].

However, for such a retrofit, often only the single part to be replaced is considered in the topology optimization and its mounting points are modeled as fixed boundary conditions. In practical application, the standalone optimization of a single part has an impact on the force distribution inside the structure to which it is connected. Furthermore, the number and the locations of joints are predefined and therefore not optimal. For instance, the FCRC bracket from figure 1.1 is bolted at ten locations with a regular bolt pattern, which was considered as fixed for the optimization. Making the locations of bolts part of the design variables in the optimization might not only provide a better bolt pattern but also influence the result of the topology optimization of the part itself. Therefore, for a full exploitation of topology optimization, it is important to optimize the parts and their connections simultaneously, i.e. to optimize an assembly as a whole.

The problem of topology optimizing parts and the layout of fasteners at the same time was tackled e.g. by Chickermane and Gea [47] in a straightforward way: As potentially force transferring joints, a grid of spring elements, acting between the nodes of two parts, is introduced. The springs are treated by the same stiffness penalization scheme as the continuum elements of the individual parts. By constraining the maximum “volume” of the springs, only the most important springs remain in the final result. However, this method does not guarantee a layout

with discrete joints. Even partial joints may be stiff enough to transfer loads sufficiently well, therefore one has no control over the number of joints remaining in the final result. Also, for higher resolution FE meshes, the spring grid gets denser and the inherent lack of control over a minimum distance between joints becomes evident: If aiming for an assembly held together by multiple separate connections, a result with just a few clustered groups of joints is obtained.

To overcome the problem of non-discrete joints, recently Thomas et al. [48] studied topology and joint layout optimization using BESO [49] and applied their method to periodic assemblies. Nevertheless, the problem of joint clustering remains, and requires several heuristic approaches and parameters to obtain the shown results. Again, a grid of potential joints with fixed spacing is used, which might lead to suboptimal results, since joints cannot be placed at any arbitrary position.

Alternative methods for combined topology and layout optimization directly use components' location variables as part of the design vector, thus allowing a flexible placement. Zhu et al. [50] applied this technique for a layout optimization of fixed-shape components embedded in a topology optimized support structure. The method was later refined by including predefined, fixed fastener layouts to model the force transfer between the components and the support structure [51]. The fasteners are linked to the support structure using coupling equations, circumventing the need for remeshing when components move. Most recently, Rakotondrainibe et al. [52] modeled joints as rigid supports by using movable boundary condition zones, whose positions are part of the optimization in a level set topology optimization framework.

Closely related to assembly optimization techniques are multi-material topology optimization approaches in which material placement and selection are optimized simultaneously. Such problems were tackled by an extended multi-phase material interpolation scheme from Sigmund and Torquato [53] and later refined by Hvejsel and Lund [54]. In these methods, multiple materials are defined either by different material properties or, in the context of fiber composites, by different layups and fiber orientations. The different material zones in the optimized result can be interpreted as different parts that need to be bonded. However, there is no distinct modeling of the bonding properties since all "components" share the same FE mesh such that connectivity is given by the mesh itself with an ideal load transfer.

In Florea et al. [55] this approach is therefore extended by an algorithm to identify these interfaces and place "joint material" in between dissimilar materials, yielding distributed bonded areas. A way to control the amount of joint material is proposed by using joint cost functions, which is then further refined by Shah et al. [56]. However, in all these multi-material methods, the count and dimensions of the individual components are not controlled because the same FE mesh is used for all components. Also, point-like connections such as bolts cannot be modeled in a simple way since the focus lies rather on large bonded surfaces.

1.3 Objective

New methods need to be developed to treat the fail-safe optimization of topology optimized parts in a computationally efficient manner. Existing methods rely either on an approach as used by Jansen et al. [35] that explicitly considers damaged configurations during the optimization but comes at an extremely high computational cost or on an indirect method like the local volume constraints by Wu et al. [44] that efficiently yields designs with multiple load paths but does not include explicit damage calculations.

This work aims at closing the gap in between these two ways by presenting algorithms that are at the same time also compatible with the basic workflow of the widely used density-based three field topology optimization approach. While most shown examples later in this work refer

to well-studied academic 2D problems to make them comparable to other published methods, also the application to at least one 3D problem is shown to underline that the methods presented in this work are also applicable in an industrial context.

The main issue of established methods for fail-safe topology optimization is, that the number of damage scenarios and therefore the computational effort scales either directly with the discretization of the FE model, as in [35], or at least with the dimensionality of the problem, as for the methods that use a sparser damage grid [36, 38].

Hence, to be usable for industrial application, it is essential to severely reduce the computational cost associated with a topology optimization towards a fail-safe structure. In this work, it is targeted to reduce the total number of simulations to be run for such an optimization to a range in the thousands (even for 3D problems), while a plain topology optimization usually requires a count in the hundreds. Only by achieving such a reduction, also high-resolution models can be optimized in a feasible amount of time. Besides that, the newly presented method should also be able to handle models with irregular meshes.

Tracing back the idea behind the fail-safe topology optimization method from [35], the huge amount of damage scenarios to consider originates from the lack of knowledge about the location, where a critical damage may appear, as in a topology optimization the final geometry of the part is yet to be found. Consequently, by using the proposed simplified rectangular damage model from [35], two main problems leading to unnecessary evaluations can be identified: At first, also regions containing void are cut and thus define an own damage scenario. Secondly, once a load path is established, multiple damage scenarios may cut only this one load path leading to redundant simulations.

The load-path-based damage model presented by the author in [37] tackles both issues at once. By using parametric damage zones that are adaptively shaped to cover all solid load paths, no calculation time is wasted on void regions and all load-bearing features of the part are included in damage consideration exactly once. While this approach decouples the number of simulations from the discretization and dimensionality, thus paving the way to fail-safe topology optimizations of large-scale structures in an industrial scale, convergence issues remain.

Therefore, in this work, the first main objective is to embed the load-path-based damage model from [37] in a different type of optimization, which is closely related to a density-based topology optimization but can handle load-path-based damage in a numerically stable way. The resulting sequential optimization procedure allows to combine the advantages of the extremely efficient approach for redundant topology optimized structures suggested in [44] with an explicit formulation of the fail-safe requirement as in [35] but with the difference, that damage is then applied to actual load paths.

Wherever multiple parts are interconnected, fail-safety aspects should also be considered on an assembly level. If an assembly is held together by joints like e.g. bolts, these joints are the load paths that are vital for the force transfer between the connected parts. Consequently, a failure of one of the joints may have a severe negative impact on the performance of the whole assembly.

To optimize assemblies with consideration of possible joint failures, these joints need to be discrete components such that again an explicit formulation of the fail-safe requirement can be used within the optimization to efficiently optimize towards a fail-safely connected assembly.

However, in existing methods for a simultaneous topology optimization of parts and their connections in an assembly, the joints in the results usually appear as distributed joint zones instead of discrete, distinct joints. This hinders the control of the desired joint count and thus also of the degree of redundancy for each connection. In revisiting e.g. the method from Chickermane and Gea [47] with today's computational power, which allows to use a much finer

FE discretization, it is found that clustered groups of joints are obtained. Also, for the recently published method of Florea et al. [55], only distributed joint areas are obtained. Therefore, in neither of both approaches, a full control over the count of distinct joint zones is possible, which consequently also makes the definition of individual joint failure scenarios cumbersome.

Hence, the second main objective of this work is to present a method for fail-safe assembly optimization in which joints are discrete, predefined features, which allows for a strict control of the total joint count. For each joint, also a detailed modeling of its local geometry and its force transfer properties is possible, such that different types of joints become distinguishable, as e.g. a bolted connection requires holes to be present in the parts to be clamped, which is not the case if spot welds are to be used instead. In including failure of joints already at the optimization stage, the topology of each individual part of the assembly and the joint layout are optimized simultaneously, while the inner loads inside the whole assembly are analyzed, even if one (or more) joints fail. The result is then an assembly connected by a fail-safe joint layout.

In further combining the method for fail-safely connected assemblies with the earlier mentioned load-path-based damage model to account for partial failure of each component, optimized assemblies can be obtained that include both fail-safe aspects and are robust with respect to a failure of interconnecting joints and possible damages that may occur to the individual parts of the assembly.

All fail-safe optimization problems discussed in this work are tackled in a worst-case sense. This means, that the optimization is driven by the damage scenario with the worst impact on the structural performance of the part or assembly, not accounting for how likely the occurrence of this particular damage is.

Probabilistic methods are not employed in this work since they require a failure probability model. The optimization is then dominated by the most probable damage state of the structure, not the one with the largest impact on the performance. When choosing realistic damage scenario probabilities, in many cases the by far most likely state is the intended, undamaged configuration (otherwise, a re-design is strongly recommended). The obtained result when using probabilistic approaches may therefore in some cases simply resemble the result of a deterministic, plain topology optimization without any damage considerations. In the works by Kriegesmann and Lüdecker [11, 57], dealing generally with probabilistic approaches for topology optimization, similar observations were made when considering random variations of material properties: Designs deviating significantly form the corresponding deterministic designs are only obtained if the scatter of the material properties is chosen artificially high.

Another aspect to consider when using probabilistic methods in fail-safe optimizations is, that, depending on the formulation of the objective, the effect of a damage for a less likely damage scenario may be amplified in favor of a reduction of damage effects on more probable failure scenarios. This leads to designs that are susceptible to a few unlikely but therefore extremely devastating failure cases (cf. [42]). A comparable effect is also visible in the deterministic approach of sparser damage grids by [36], where failure scenarios at specific locations are disregarded to save computational cost. In removing these scenarios from the optimization, the most extreme failures are shifted to these non-considered regions in order to be “hidden” from the evaluation.

Considering both aforementioned aspects, this work concentrates on computationally efficient implementations of the conservative worst-case approach for fail-safe optimization to comply with the requirements in CS-25 [14] that demand a multiple load path construction. However, the presented methods may always be embedded in probabilistic optimization formulations to adapt them to other use cases.

1.4 Outline

In section 2, the basic theory of the density-based three field topology optimization is recapitulated. Additionally, modifications to a plain topology optimization that are relevant for this work are presented and example designs for two widely studied academic problems are given.

Since most of the optimization approaches presented in this work involve aggregation of objectives or constraints, the topic of aggregation is separately discussed in section 3. Besides studying the advantages and disadvantages of known aggregation functions, also a newly derived aggregation scheme is presented.

Section 4 is dedicated to fail-safe topology optimization considering partial damage. Starting from the simplified rectangular damage model from [35], methods to reduce the involved computational cost are evaluated, namely the method presented in [36] and the "threshold method" by the author [37]. In difference to the preceding publication [37], an extended examination with more examples and higher resolution models is presented. Finally, the load-path-based damage model [37] by the author is introduced, which is then used throughout the rest of this work.

Section 5 adapts the standard three field topology workflow for a use as "density-based shape optimization". The new optimization method is at first introduced as a general optimization algorithm and then the specific application to fail-safe optimizations employing the load-path-based damage model is given. Therein, the presented density-based shape optimization is used as a tool for a numerically stable, two-stage procedure that rapidly yields results for fail-safe optimizations involving partial damage. Numerical examples for 2D and 3D problems are given. A discussion of the total computational cost in section 5.4 sets the proposed method in comparison to known methods for fail-safe optimizations and reveals that computational savings of at least three orders of magnitude are achieved for the shown examples. Compared to the preceding publication of the author in [58] concerning the proposed optimization method, higher resolution models are used, additional examples are considered, and even higher computational savings are achieved. Additionally, a further reduction of computational cost for compliance-based fail-safe optimizations is proposed and assessed in this work.

The optimization of assemblies is discussed in section 6. At first, known methods relying on pure topology optimization algorithms are revisited with high-resolution FE models. Then, a new approach for simultaneous topology and joint layout optimization is presented that involves continuously movable joints and non-design space zones. As in the preceding publication of the author [59], the proposed assembly optimization method is further augmented by damage aspects in explicitly considering failure of single or multiple joints, which is demonstrated for 2D and 3D examples. In this work, additionally an implicit way to obtain joint layouts with increased damage tolerance is presented, in which interface forces are more evenly distributed across all joints. As a further novelty, the load-path-based damage model from section 5 is embedded in the proposed assembly optimization method from section 6 to yield fail-safe assemblies, which is discussed in section 7.

A conclusion, recapitulating the most important findings of this work, and an outlook, discussing possible future extensions, are given in section 8.

Appendices A – C contain additional theory, analytical derivations, and numerical parameter studies for the interested reader, which underline important aspects of the presented methods in this work. Appendices D and E are dedicated to the derivatives required for an implementation of the proposed methods.

2 Basic theory of topology optimization

This work focuses on linear, static mechanical systems, where external loads are applied and the displacements are the unknowns to be calculated using the finite element method (FEM).

Various efficient implementations for topology optimization in a continuum setting exist in the literature. Starting from the 99-line Matlab code `top99` for 2D compliance topology optimization [60], an 88-line code `top88` [61] offering a full design variable filtering step was made available. The most recent compact codes for 2D and 3D compliance minimization `top99neo` and `top3D125` [62] also include a projection step and therefore follow the three field topology optimization workflow presented in the following sections, on which also this work is based on.

The mathematical formulation of the topology optimization problem is presented in section 2.1. Section 2.2 outlines an iterative, gradient-based solution procedure and breaks down the involved numerical simulation process and the optimization algorithm itself into its substeps. These substeps are then discussed in detail in sections 2.3 – 2.5, covering the topics of data processing yielding three fields of variables, objective and constraint functions, and parameters for the optimization algorithm used. Section 2.6 presents modifications to the plain topology optimization task, allowing to prescribe desired properties of the optimized results. Results for two academic examples obtained by the presented methods are shown in section 2.7.

2.1 Optimization problem formulation

The basic constrained optimization problem for a topology optimization (TO) task states [63,64]:

$$\min_{\boldsymbol{\varrho}} \Phi(\boldsymbol{\varrho}) \tag{2.1a}$$

$$\text{s.t.} \quad h_i(\boldsymbol{\varrho}) \leq 0, \quad \forall i \tag{2.1b}$$

$$0 \leq \varrho_j \leq 1, \quad j = 1, \ldots, n_e \tag{2.1c}$$

$$\boldsymbol{K}(\boldsymbol{\varrho})\boldsymbol{u} = \boldsymbol{f} \tag{2.1d}$$

The function Φ is the objective to be minimized by altering the entries of the design vector $\boldsymbol{\varrho}$. Additionally, several inequality constraints h_i may be present, limiting the feasible domain of the design variables. At least one of the objective or constraint functions is also dependent on the displacement vector $\boldsymbol{u}$, being the solution of equation (2.1d) for a linear elasticity problem.

Typically, each of the n_e elements of the FE model is assigned an independent design variable (DV) ϱ_j.

The idea behind the DVs is, that a value of 0 or 1 means, that the element is considered as being part of the void or solid phase, respectively. However, in current TO implementations, the DVs are processed by a filtering and a projection step, before they are used as an indicator for an element being solid or void. These steps are further explained in section 2.3.

Additionally, since the mathematical treatment of an optimization problem with binary variables is cumbersome, in equation (2.1c), the problem is already considered in a relaxed form, with continuous DVs bounded by an interval $\varrho_j \in [0, 1]$. An interpolation of intermediate values therefore becomes necessary, which is discussed in section 2.3.4.

2.2 Optimization methodology

The nonlinear optimization problem presented in section 2.1 is solved iteratively through a gradient-based optimization algorithm. A diagram of the main steps of the optimization

algorithm is shown in figure 2.1. The actually used optimization algorithm is further discussed in section 2.5.

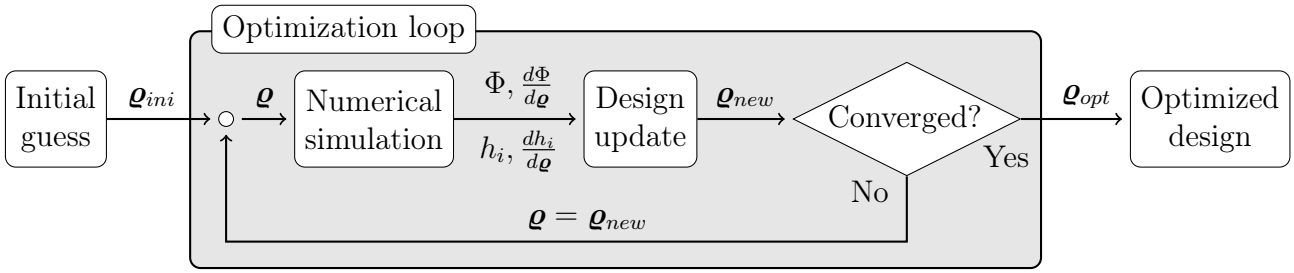


Figure 2.1 – The main steps of an iterative optimization algorithm.

The optimization is started with an initial guess for the DVs. A numerical simulation is performed to obtain the values of the objective Φ and constraint functions h_i. Additionally, their gradients with respect to the DVs are calculated. Based on these information, a new set of DVs is generated in a design update step with the goal to obtain an even lower objective value while moving in the feasible domain of the solution space. At the end of each iteration, convergence criteria are checked. In the simplest case, convergence is judged e.g. by the maximum change in the DVs for two successive iterations [60]. If the convergence criteria are not met, another optimization iteration is performed, but this time with the new DVs as the baseline. The optimization finishes if convergence is achieved yielding a set of DVs describing an optimized design.

The focus of this work lies in implementing new methods for the "numerical simulation" block from figure 2.1. Also, it is assumed that the simulation step consumes the majority of processing time for the optimizations considered. For further analysis, figure 2.2 shows a breakdown of the numerical simulation process, that is repeated for every optimization iteration.

The DVs used for the TO may undergo several processing steps (steps 1 to n in figure 2.2) until a set of processed variables is obtained. Each processing step may also require the definition of additional parameters. The processed variables describe either a single or in some cases m distinct realizations ($\hat{\boldsymbol{\varrho}}_1, \ldots, \hat{\boldsymbol{\varrho}}_m$ in figure 2.2) of a design.

Figure 2.2 shows an exemplary case, where the evaluation of the objective Φ and its gradients involves the calculation of the structural performance of the underlying design. Therein, the processed variables are used for the assembly of the global system matrices. For each of the m realizations, an own FE analysis is conducted. Afterward, an aggregation step may be needed to obtain a combined objective function. In contrast to that, the constraint functions h_i in figure 2.2 are considered to be simple analytic functions of the processed variables, therefore their values and gradients can be directly computed, without additional FE analyses. In cases where a large number of similar constraints are present, it may be favorable to aggregate them into a joint constraint. Aggregation strategies for objectives and constraints are further discussed in section 3.

Throughout this work, the individual steps seen in figure 2.2 are adapted to address various optimization problems. Therefore, much simpler diagrams, like the one seen in figure 2.3, are used to describe the numerical simulation process run for every optimization iteration. Therein, the DVs will be shown in red, the processed variables used for the assembly of the system matrices in green, and additional input parameters in blue. If several realizations of a design need to be simulated, also several processed variable vectors are shown. Their count equals the number of FE analyses conducted per optimization iteration. The topic of aggregation is addressed separately in the corresponding text sections for each use case.

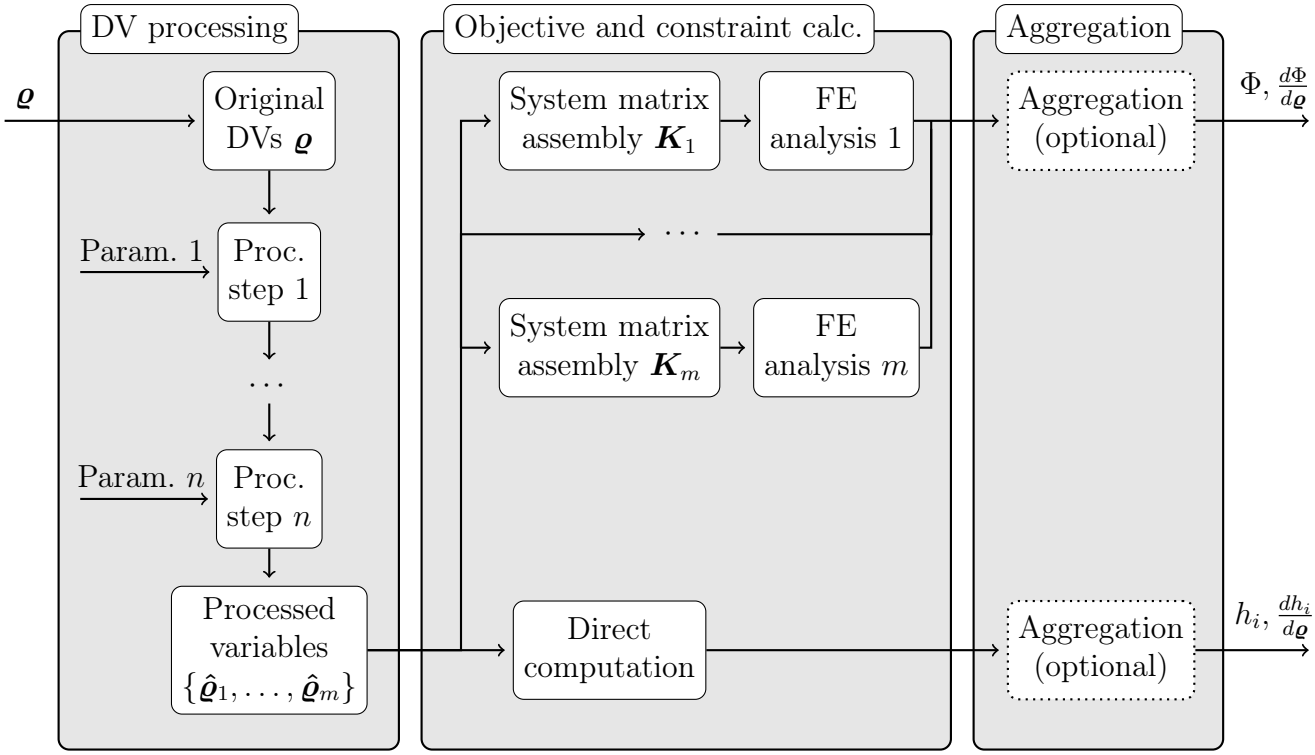


Figure 2.2 – Breakdown of the numerical simulation process.

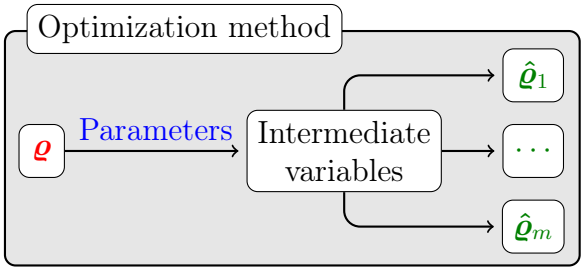


Figure 2.3 – Simplified numerical simulation process diagram with the DVs (red), parameters (blue), and processed variables (green) that serve as input for assembly of the system matrices.

2.3 Three field topology optimization

The widely used workflow for topology optimization including three fields is adopted in this work. The three fields are [26]:

1. The design variables $\boldsymbol{\varrho}$,
2. the filtered variables $\tilde{\boldsymbol{\varrho}}$,
3. and the projected variables $\bar{\boldsymbol{\varrho}}$.

The individual variables in the three fields are all bounded in the interval of $[0, 1]$.

2.3.1 Filtering of the design variables

The DVs $\boldsymbol{\varrho}$ of the first field are mapped to filtered variables $\tilde{\boldsymbol{\varrho}}$ by a mesh independency filter [29, 30]. The filtered variables $\tilde{\boldsymbol{\varrho}}$ form the second field and are calculated as a weighted mean of the DVs of surrounding elements within a filter radius r:

$$\tilde{\varrho}_i = \frac{\sum_j w(\boldsymbol{c}^j - \boldsymbol{c}^i) v_j \varrho_j}{\sum_j w(\boldsymbol{c}^j - \boldsymbol{c}^i) v_j} \tag{2.2a}$$

$$w(\boldsymbol{x'}) := \max(r - \|\boldsymbol{x'}\|_2, 0) \tag{2.2b}$$

Here, v_j are the element volumes, $\boldsymbol{c}^i$ and $\boldsymbol{c}^j$ are the locations of the element centers and w is a conic weighting function[2] dependent on the filter radius r.

The filter introduces a spatial correlation of the filtered variables, suppressing abrupt changes from 0 to 1 or vice versa between neighboring elements if the radius r is chosen sufficiently large. Therefore, filtering is an efficient way to suppress checkerboard patterns, that would occur otherwise if material penalization (see section 2.3.4 and Appendix C.2) is used. Additionally, filtering leads to mesh independence, since the final topology is now dominated by the choice of the filter radius r instead of the element size of the underlying FE mesh [29].

Due to the averaging effect of the filter, the fastest transition from void to a solid feature in the design space takes at least a distance of r, within which filtered variables with intermediate values between 0 and 1 exist (see figure C.2c in Appendix C.2).

Filtering of boundary regions requires special treatment to avoid the introduction of artificial symmetry effects at free boundaries [65,66]. Appendix C.1 shows the approach used in this work to deal with this problem.

The sensitivities $\frac{\partial \tilde{\varrho}_i}{\partial \varrho_j}$ of the filter equation are given in Appendix D.3.

2.3.2 Projection of the filtered variables

The projected variables $\bar{\boldsymbol{\varrho}}$ are the third field in the presented approach. They are obtained from the filtered variables $\tilde{\boldsymbol{\varrho}}$ via a smooth approximation to the Heaviside step-function with a variable steepness parameter β and a threshold parameter η [31]:

$$\bar{\varrho}_i = \frac{\tanh(\beta\eta) + \tanh\left(\beta(\tilde{\varrho}_i - \eta)\right)}{\tanh(\beta\eta) + \tanh\left(\beta(1 - \eta)\right)} \tag{2.3}$$

The effect of the projection step is to allow for an increasingly sharper transition of void to solid when the parameter β is increased. Usually, for plain topology optimization, the threshold parameter is set to $\eta = 0.5$.

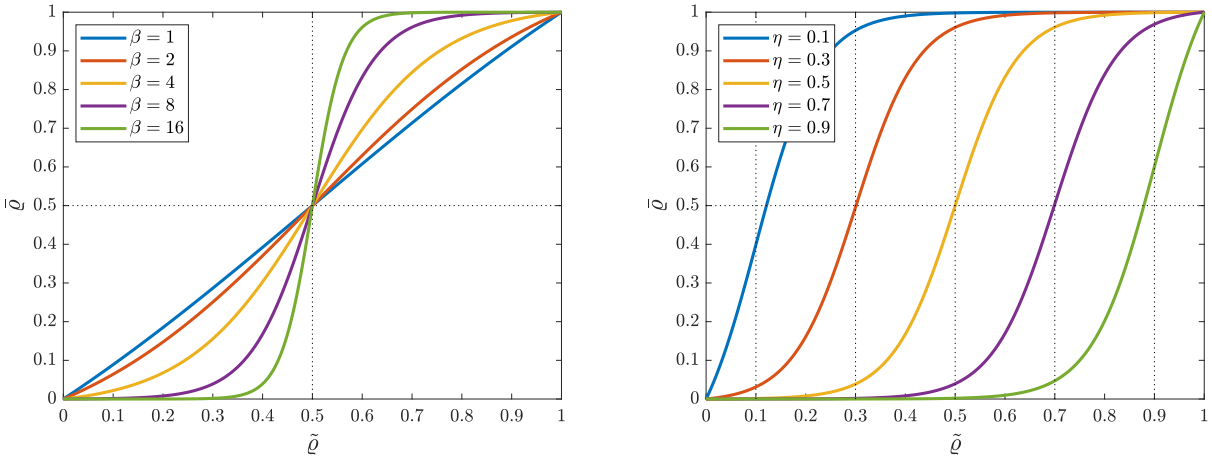
The projection characteristics for different β values and $\eta = 0.5$ is shown in figure 2.4a. Varying the threshold η shifts the range of the input values being projected towards 0 (meaning void) or 1 (solid). Figure 2.4b shows the projection curves for different η values for $\beta = 8$. The turning point of the curves is located at $\tilde{\varrho} = \eta$.

The sensitivities $\frac{\partial \bar{\varrho}_i}{\partial \tilde{\varrho}_i}$ of the projection equation are given in Appendix D.4.

2.3.3 Parameters in the three field topology optimization approach

Figure 2.5 summarizes how the three fields are related to each other and which parameters have an influence on the individual mapping steps. The DVs are shown in red, the predefined parameters are blue and the physical densities, used for evaluating the stiffness of the individual elements (see section 2.3.4), are shown in green.

[2] In general, w is a weighting function dependent on a characteristic length r. Here, a conic weighting function is used for w, where r corresponds to the radius of the support of the function, therefore r is referred to as the filter radius. There exist other weighting functions like e.g. a constant or quasi-Gaussian bell function, however, no major advantages can be seen for typical TO application in using these alternatives, according to [65].



(a) Projection with $\eta = 0.5$ and varying β **(b)** Projection with $\beta = 8$ and varying η

Figure 2.4 – Projection functions for different β and η values.

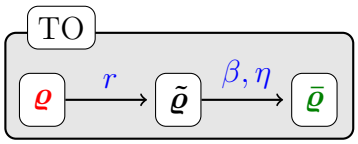


Figure 2.5 – Relation of variables for the three field TO approach.

For the standard topology optimization approach, which is in the case of compliance minimization for example implemented in the 99-line Matlab code `top99neo` for 2D and the 125-line code `top3D125` for 3D [62], the projected variables $\bar{\boldsymbol{\varrho}}$ are the physical pseudo-densities used for the stiffness evaluation in the FE model. The DVs in $\boldsymbol{\varrho}$ themselves have no physical meaning in the discussed three field approach.

2.3.4 Assembly of the global system matrix and material penalization schemes

In this work, only linear elastic, isotropic materials are considered. To assemble the global stiffness matrix $\boldsymbol{K}$, the individual elements' stiffness matrices $\boldsymbol{k}_i$ are needed ("A" is the assembly operator):

$$\boldsymbol{K} = \bigwedge_{1}^{n_e} \boldsymbol{k}_i(\bar{\varrho}_i) = \bigwedge_{1}^{n_e} E_i(\bar{\varrho}_i)\boldsymbol{k}_0 \tag{2.4}$$

The element's stiffness matrix $\boldsymbol{k}_i$ for element i is calculated based on an effective Young's modulus E_i dependent on the projected variable $\bar{\varrho}_i$ for this element[3]. Since the pseudo-densities are continuous in $\bar{\varrho}_i \in [0, 1]$, an interpolation scheme for intermediate values is needed:

$$E_i = E_{min} + \kappa(\bar{\varrho}_i)(E_0 - E_{min}) \tag{2.5}$$

[3]For the linear elastic, isotropic case, the Young's modulus E_i can be dragged out of the element's stiffness matrix, making $\boldsymbol{k}_i$ a scaled version of a unitary elemental stiffness matrix $\boldsymbol{k}_0$, that was calculated with $E = 1$.

Here, κ is the interpolation function, E_0 and E_{min} are the Young's moduli of the solid and void phase, with $E_{min} \ll E_0$. To counter singular systems, also for the void phase some small finite stiffness has to be defined by choosing a nonzero positive value for E_{min}.

A common choice is to use a power law [23] as interpolation function, which is also known as the SIMP (Solid Isotropic Material with Penalization) approach:

$$\kappa(\bar{\varrho}_i) = \bar{\varrho}_i^p \tag{2.6}$$

The alternative RAMP (Rational Approximation of Material Properties) approach uses a rational interpolation function [67]:

$$\kappa(\bar{\varrho}_i) = \frac{\bar{\varrho}_i}{1 + q(1 - \bar{\varrho}_i)} \tag{2.7}$$

The parameters p in equation (2.6) and q in equation (2.7) are used to penalize intermediate densities. If $p > 1$ or $q > 0$ is chosen, intermediate densities are less attractive, since the ratio of the effective stiffness E_i/E_0 is smaller than the relative cost in volume of these elements. Therefore, optimizations, in which the total volume is constrained or minimized, converge to a binary density distribution.

The penalization effect is stronger with increased parameter values for p or q, respectively. Figure 2.6 shows the penalization functions for varying parameters p and q, compared to a linear interpolation and the Hashin-Shtrikman upper bound for a two-phase composite at a Poisson's ratio of $\nu = 1/3$ (for more details see [67, 68]). Figure 2.6 uses $E_{min} = 0.1E_0$ for visualization purposes. However, all numerical examples in this work are calculated with $E_{min} = 10^{-9}E_0$.

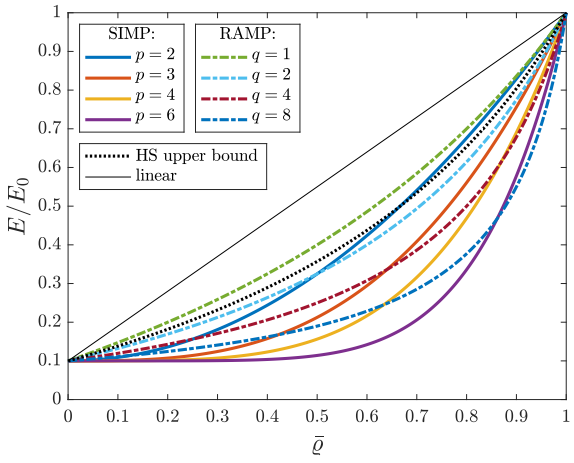


Figure 2.6 – Material penalization schemes SIMP and RAMP for different parameters and $E_{min} = 0.1E_0$.

The derivatives $\frac{\partial \boldsymbol{K}}{\partial \bar{\varrho}_i}$ of the global stiffness matrix $\boldsymbol{K}$ with respect to the projected variables $\bar{\varrho}_i$ dependent on the chosen material penalization scheme are given in Appendix D.5.

Appendix C.2 demonstrates the effect of penalization, filtering, and projection for the example of a cantilever beam compliance minimization with a global volume constraint.

2.4 Objective and constraint functions

In this section, design response functions for mechanical optimization problems being relevant for this work are presented, together with their derivatives. Based on the commonly used optimization task of compliance minimization under a prescribed maximum allowed volume, the design responses are further classified into being an objective or a constraint function. However, this classification is not unambiguous and depends on the goal of the optimization. For the material penalization scheme from section 2.3.4 to work, at least one of the objectives or constraints needs to be a volume function.

2.4.1 Compliance objective

The compliance as objective is defined as the dot product of the displacements and the external force vector:

$$\Phi = c(\boldsymbol{\varrho}) = \boldsymbol{f}^T \boldsymbol{u} \tag{2.8}$$

For the linear static case, the compliance is twice the strain energy stored in the loaded structure. To obtain the derivative with respect to the projected variables, the adjoint method is used (see Appendix D.2). The adjoint vector $\boldsymbol{\lambda}$ is obtained by:

$$\boldsymbol{K}\boldsymbol{\lambda} = -\boldsymbol{f} \tag{2.9a}$$
$$\boldsymbol{\lambda} = -\boldsymbol{u} \tag{2.9b}$$

Considering compliance in the linear static case, the displacements can be reused as the solution to the adjoint system, such that no additional calculation is needed to obtain the gradients $\frac{dc}{d\bar{\varrho}_i}$:

$$\frac{dc}{d\bar{\varrho}_i} = -\boldsymbol{u}^T \frac{d\boldsymbol{K}}{d\bar{\varrho}_i}\boldsymbol{u} \tag{2.10}$$

The gradient with respect to the DVs is obtained by the chain rule (see Appendix D.1):

$$\frac{dc}{d\varrho_i} = \sum_j \frac{\partial c}{\partial \bar{\varrho}_j}\frac{\partial \bar{\varrho}_j}{\partial \tilde{\varrho}_j}\frac{\partial \tilde{\varrho}_j}{\partial \varrho_i} \tag{2.11}$$

The chain length is three, corresponding to the three fields shown in figure 2.5.

2.4.2 Global volume constraint

The part's global volume fraction ω may be constrained to a value k_g, serving as an upper bound. The constraint equation h then states:

$$h = \omega(\boldsymbol{\varrho}) - k_g \leq 0 \tag{2.12}$$

The global volume fraction ω is the ratio of the part's current volume V in relation to the full volume of the design space V_0. The part's volume V is the sum of the element volumes weighted by the projected densities:

$$\omega = \frac{V}{V_0} = \frac{\sum_i \bar{\varrho}_i v_i}{\sum_i v_i} \tag{2.13}$$

Since for the linear case and small displacements the calculation of the volume fraction does not involve the displacement vector $\boldsymbol{u}$, direct differentiation is performed to obtain the gradients of h with relation to the projected variables:

$$\frac{dh}{d\bar{\varrho}_i} = \frac{v_i}{\sum_j v_j} \tag{2.14}$$

The gradient with respect to the DVs is then obtained by the chain rule (see Appendix D.1):

$$\frac{dh}{d\varrho_i} = \sum_j \frac{\partial h}{\partial \bar{\varrho}_j} \frac{\partial \bar{\varrho}_j}{\partial \tilde{\varrho}_j} \frac{\partial \tilde{\varrho}_j}{\partial \varrho_i} \tag{2.15}$$

2.4.3 Local volume constraints

A local volume constraint is similar to the global volume constraint but acts on a (much) smaller control set. Local volume constraints can be used e.g. for limiting the maximum member size [69, 70] and to enforce "porous" designs with redundant load paths [44, 71]. More details about these applications are given in sections 2.6.2 and 2.6.3.

With a local volume constraint h_i, the volume fraction of elements that have their centers within a neighborhood $\mathbb{N}_i$ of radius R around the center point of element i can be constrained to a maximum value k_l: [44]

$$h_i = \omega_i(\boldsymbol{\varrho}) - k_l \leq 0 \tag{2.16a}$$

$$\omega_i = \frac{\sum_{j \in \mathbb{N}_i} \bar{\varrho}_j v_j}{\sum_{j \in \mathbb{N}_i} v_j} \tag{2.16b}$$

$$\mathbb{N}_i = \{j \mid \|\boldsymbol{c}_j - \boldsymbol{c}_i\|_2 \leq R\} \tag{2.16c}$$

For obtaining the gradients, the equations for the global volume constraint from section 2.4.2 can be used if the set of indices, over which the summations are performed, is limited to $\mathbb{N}_i$.

Using local volume constraints gives rise to a large number of constraints since typically one local volume constraint per finite element is employed. However, these n_e individual constraints may be aggregated to a single constraint through aggregation functions (see section 3).

Using aggregation functions reduces computational cost for the optimization algorithm, however, the aggregation strategies presented in this work involve a trade-off between accuracy and numerical stability. This aspect is thoroughly discussed in section 3, where also an alternative aggregation scheme is derived for this matter.

2.5 Choice of the optimization algorithm and parameters

Basically, any gradient-based algorithm capable of solving nonlinear problems with a large number of DVs can be used to solve the studied optimization problem from equations (2.1). Non-gradient-based algorithms like e.g. genetic algorithms are not recommended for problems with many DVs, since they would require an immense number of function evaluations to achieve convergence [72].

In this work, the "Method of moving asymptotes" (MMA) [73, 74], which is described as a flexible and general method for structural optimization, is used as a gradient-based optimization algorithm. The MMA can solve optimization problems involving n_c constraints of the following form[4]:

$$\min_{\boldsymbol{\varrho},\boldsymbol{y},z} \Phi(\boldsymbol{\varrho}) + a_0 z + \sum_{i=1}^{n_c} \left(c_i y_i + \frac{1}{2} d_i y_i^2 \right) \tag{2.17a}$$

$$\text{s.t.} \quad h_i(\boldsymbol{\varrho}) - a_i z - y_i \leq 0, \quad i = 1, \ldots, n_c \tag{2.17b}$$

$$0 \leq \varrho_j \leq 1, \quad j = 1, \ldots, n_e \tag{2.17c}$$

$$y_i \geq 0, \quad \forall i \tag{2.17d}$$

$$z \geq 0 \tag{2.17e}$$

To solve the problem from equations (2.1), involving a single scalar objective function Φ, the recommended values for the coefficients a_0, a_i, c_i and d_i are [74]:

$$\left.\begin{array}{ll} a_0 & = 1 \\ a_i & = 0 \\ c_i & \gg 0 \\ d_i & = 1 \end{array}\right\} i = 1, \ldots, n_c \tag{2.18}$$

The auxiliary variables $\boldsymbol{y}$ and z from equations (2.17) allow a broad application of the MMA to different types of optimization problems [74]. For standard TO problems, these variables are usually not relevant. However, in "worst-case" optimization scenarios, where the maximum over several objective functions values is to be minimized, a suitable choice of the coefficients from equation (2.18) can be made such that the variable z is used as a slack variable to approach the underlying min-max-problem (for details, refer to [64, 74]). Nevertheless, in this work, min-max-problems are tackled by aggregation strategies that are discussed in detail in section 3. One reason for this is, that by using the slack variable approach, the complexity of the optimization task increases since the objective functions are moved over to the constraints. This increases the computational time of the corresponding dual problem. Opposed to that, aggregation functions actually decrease the complexity of the optimization task, since several functions are combined into a joint function. Another reason is, that aggregation functions involve an own tuning parameter, such that the contribution of each single objective value to the joint objective, and thus for the course of the optimization, is transparent at any time (see section 3.5.1), in contrast to the case where the prioritization is being handled internally by the MMA when using the slack variable.

[4] Compared to [74], the notation in equations (2.17) is slightly adapted to the symbols used in this work.

2.5.1 Initial values

For a nonlinear optimization problem, several local optima may exist. Therefore it is advisable to try different starting values when using a gradient-based optimization method.

For volume constrained TO, the DVs are often initialized with a uniform value of the desired volume fraction [60–62]. Using different values for a uniform starting field, e.g. by using a prefactor, may already lead to different topologies as local minima of the optimization problem (see Appendix C.4 and [27]). Therefore, for most examples in this work, different uniform starting fields have been considered to obtain the shown results.

Additionally, depending on the parameters for the projection step (see section 2.3.2) and the chosen volume fraction, not using a prefactor might in some cases yield an initial guess that is in the infeasible domain of the volume constraint. This is due to the volume being calculated based on the projected variables in the three field TO approach and not directly on the DVs. Therefore, a prefactor for compensation might be needed anyway.

Alternatively, randomized starting fields could be considered, but this would lead to (suboptimal) local minima with non-symmetric designs for the shown examples involving symmetric models and loading. Therefore, no random starting fields are used in this work.

2.5.2 Convergence criterion

This work adopts the stopping criteria from [60–62], where the optimization terminates if the maximum change in the DVs is below a certain threshold or if a defined maximum number of optimization iterations is reached.

For most of the shown examples, the maximum number of iterations is set to 200, if not stated otherwise. The threshold for the maximum absolute change of the DVs has a deliberately low value of 10^{-6}, such that all optimizations ended due to reaching the maximum number of iterations. However, the convergence behavior was monitored and the designs for the newly presented methods had stabilized usually within the first 50 iterations, such that after that no gross changes in topologies were observed.

2.5.3 Fixed parameters and variable continuation

The projection step of the three field TO approach (see section 2.3.2) may lead to numerical instabilities if the steepness parameter β is chosen too high from the start on. One approach to deal with this problem is to initialize β with a low value and double its value e.g. every 50 iterations until the desired maximum value is reached [31, 75, 76]. Other parameters are left unchanged during a plain TO. Table C.1 in Appendix C.6 lists the settings that are used for the academic examples shown in this work.

2.6 Modifications to plain topology optimization

Additional aspects may be added to the plain three field TO workflow, to gain control over the properties of the optimized designs.

Examples discussed in this section include the “robust” approach to handle manufacturing uncertainties, enforcing redundant designs by using local volume constraints, as well as minimum and maximum feature size control.

Other possible aspects, like e.g. manufacturing constraints (overhang angle control, milling or casting constraints, etc.) are beyond the scope of this work and are not treated in this section.

2.6.1 Three designs for the robust topology optimization approach

A manufacturing tolerant approach for topology optimization was introduced by Sigmund [75] and later refined to the "robust"[5] approach by Wang et al. [31]. Most recently, Aage et al. [78] have applied this method also to level-set-based large-scale TO problems. The motivation behind these methods is to include manufacturing errors that may result in uniformly too thin or too thick features, deviating from the desired design, already at the optimization stage. Thereby, a three projection approach is used: By altering the value of the projection threshold η uniformly (see section 2.3.2), eroded or dilated designs can be obtained, while the nominal design acts as the desired blueprint design. The effect of a uniform variation of η is demonstrated in Appendix C.3.

With the three threshold values η_e, η_0, and η_d, yielding the projected variables $\bar{\boldsymbol{\varrho}}_e$, $\bar{\boldsymbol{\varrho}}$, and $\bar{\boldsymbol{\varrho}}_d$ for the eroded, nominal, and dilated design (see figure 2.7), the objective is to minimize the worst-case configuration [31]:

$$\min_{\boldsymbol{\varrho}} \max\left(\Phi(\bar{\boldsymbol{\varrho}}_e), \Phi(\bar{\boldsymbol{\varrho}}), \Phi(\bar{\boldsymbol{\varrho}}_d)\right) \tag{2.19}$$

In the general case, the objective has to be evaluated for all three configurations and the optimization should be driven by the worst-case configuration. Since the max-function is not differentiable and the type of optimization is a worst-case optimization the objective function is approximated by means of the KS aggregation function A_{KS} (for a discussion, see section 3.5.1). Using KS aggregation, the objective function from equation (2.19) is altered to:

$$\min_{\boldsymbol{\varrho}} A_{KS}\left(\Phi(\bar{\boldsymbol{\varrho}}_e), \Phi(\bar{\boldsymbol{\varrho}}), \Phi(\bar{\boldsymbol{\varrho}}_d)\right) \tag{2.20}$$

Writing out the KS function with its aggregation parameter p, the whole optimization task for the robust approach considering e.g. a global volume constraint then states:

$$\min_{\boldsymbol{\varrho}} \frac{1}{p} \log\left(\mathrm{e}^{p\Phi(\bar{\boldsymbol{\varrho}}_e)} + \mathrm{e}^{p\Phi(\bar{\boldsymbol{\varrho}})} + \mathrm{e}^{p\Phi(\bar{\boldsymbol{\varrho}}_d)}\right) \tag{2.21a}$$

$$\text{s.t.} \quad \frac{V(\bar{\boldsymbol{\varrho}}_d)}{V_0} - k_g \leq 0 \tag{2.21b}$$

$$0 \leq \varrho_j \leq 1, \quad j = 1, \ldots, n_e \tag{2.21c}$$

$$\boldsymbol{K}\left(\bar{\boldsymbol{\varrho}}_e\right)\boldsymbol{u}_e = \boldsymbol{f} \tag{2.21d}$$

$$\boldsymbol{K}\left(\bar{\boldsymbol{\varrho}}\right)\boldsymbol{u} = \boldsymbol{f} \tag{2.21e}$$

$$\boldsymbol{K}\left(\bar{\boldsymbol{\varrho}}_d\right)\boldsymbol{u}_d = \boldsymbol{f} \tag{2.21f}$$

The simplified data processing diagram for this optimization task is given in figure 2.7.

Equation (2.21b) already considers that the worst case for the volume of a part is always achieved for the dilated design.

For the special case of compliance as objective function, the worst-case objective value will always be dictated by the eroded design, since any removal of material will decrease the stiffness

[5]The "robust" approach for TO discussed here is a deterministic optimization strategy and is not to be confused with robust topology optimization (RTO), which is a type of robust design optimization (RDO). RDO methods embed probabilistic analyses in an optimization (see e.g. [11, 57, 77]). However, this work considers only deterministic optimizations, therefore, the term "robust" in conjunction with TO refers to the mentioned method involving three projection thresholds throughout the rest of this work.

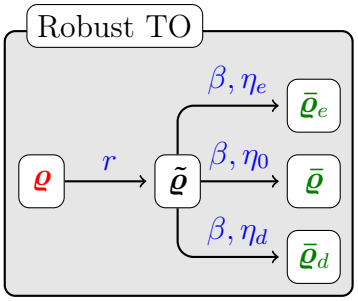


Figure 2.7 – Relation of variables for the robust TO approach.

of the structure. Therefore, for compliance minimization problems, only one FE calculation has to be done for the eroded design, instead of a total of three analyses, such that also aggregation becomes obsolete. The task for compliance minimization with a global volume constraint then simplifies to:

$$\min_{\boldsymbol{\varrho}} \boldsymbol{f}^T \boldsymbol{u}_e \tag{2.22a}$$

$$\text{s.t.} \quad \frac{V(\bar{\boldsymbol{\varrho}}_d)}{V_0} - k_g \leq 0 \tag{2.22b}$$

$$0 \leq \varrho_j \leq 1, \quad j = 1, \ldots, n_e \tag{2.22c}$$

$$\boldsymbol{K}\left(\bar{\boldsymbol{\varrho}}_e\right) \boldsymbol{u}_e = \boldsymbol{f} \tag{2.22d}$$

In cases where it is rather desired to constrain the volume of the nominal design instead of the dilated design, e.g. since a slightly larger volume due to manufacturing errors is not considered as critical, the volume constraint should still be formulated with respect to the dilated design using the adaptive strategy from [31]. This facilitates the penalization of thin members in the design and thus helps to obtain results with a constant topology across the three realizations. However, some problems are so sensitive, that even a second filtering and projection step has to be applied to further penalize thin members, as shown in [79].

The effect of the robust approach on the obtained design is twofold: On the one hand, small features under a certain minimum size are strongly penalized and therefore suppressed (see section 2.6.3), on the other hand, an overall simpler topology is obtained since implicitly the surface area of the part is minimized because the erosion acts on the boundaries only. Therefore, one thick feature is less affected by erosion, than two thinner separate features of the same volume.

Designs, optimized using the robust approach, are shown in figures 2.8c, 2.8f, 2.9c, and 2.9f in the examples section 2.7.

2.6.2 Obtaining redundant structures

Redundant structures can be cheaply obtained with the local volume constraints from equations (2.16), presented in [44]. Limiting the volume fraction inside local neighborhoods around each element, and thus enforcing a certain amount of void to be present therein, has the effect of splitting overly thick features of the design into separate, thinner features. Altering the size of the local neighborhoods allows to tune the degree of redundancy of the resulting design.

Using a relatively large radius of the control spheres and a medium local volume fraction, "porous" designs with clearly separated redundant load paths are obtained since the local volume constraints can then be seen as a combination of a maximum member size constraint and a

minimum separation constraint (for details, see [70]). Smaller radii yield globally more and finer solid features. Spatially varying local porosity control is also possible, as described in [80].

Structures with enforced redundancy are shown e.g. in figures 2.8e, 2.8f, 2.9e, and 2.9f in the examples section 2.7. Such designs offer increased robustness to partial failure (see [44], section 4.2.1, and section 5.3) and are therefore good candidates for a preliminary design for an optimization towards a fail-safe design using the method presented in section 5.

An alternative, implicit way to satisfy local volume constraints is given in [71], where the local volume fractions are considered already inside the material penalization scheme, such that no explicit additional constraints are needed. However, this approach does not follow the standard three field approach for TO and is therefore not considered in this work.

2.6.3 Size control

Above-mentioned modifications applicable to a TO can also be used as means to control the minimum and maximum desired size of structural members evolving during TO.

This section focuses on minimum size control by using the "robust" approach and maximum size control via local volume constraints since these approaches fit into the presented TO framework.

Several other approaches for feature size control in TO are reported in the literature, relying e.g. on alternative projection methods [69,76], real geometric constraints [81], explicit description of the geometry [82,83], or frequency domain transformations [84], which however is beyond the scope of this work.

Maximum member size control Approaches for controlling the maximum member size of solid features include the enforcement of a minimal amount of void [69, 70, 85] or an equivalent formulation involving the limitation of the maximum allowable amount of soild [44, 71] inside local spherical control regions. These approaches are covered by the local volume constraint from section 2.4.3. For this use case, the local volume fraction k_l is chosen relatively high (just below the maximum value of 1) together with a relatively low control region radius R, when compared to a use aiming at enforcing a redundant structure (cf. section 2.6.2).

In this work, the maximum local volume fraction is set to $k_l = 0.95$ and the value of $2R$ then approximates the upper bound of the maximum width of all solid features in the design domain. Maximum member size controlled designs are shown in figures 2.8d and 2.9d in section 2.7.

Despite being based on the same equations, the local volume constraints are treated differently in the literature, depending on their use as a means to obtain redundant designs or to limit the maximum solid feature size: Although the local volume constraints are aggregated by the same p-mean aggregation function (A_{pm}, discussed in section 3.2.2), the used aggregation parameter p is set to a moderate value of 16 (as in [44]) for the first and up to a very high value of 300 (as in [85]) for the latter case, since otherwise the desired effect is not achieved.

However, in this work, a specifically tailored alternative aggregation function A_{uni} is derived in section 3.4.3 that allows the use of a moderate aggregation parameter value of $p = 16$ for both above-mentioned use cases, due to its better approximation quality. As a result, a numerically more stable behavior of the constraint is achieved. Undesired undercuts at intersections of solid members are reported in maximum member size controlled TO problems using very high aggregation parameter values (see [70,83]), which is also remedied by using the newly introduced aggregation function A_{uni} with lower p values.

Minimum member size control A minimum member size control is implicitly included in the "robust" approach for TO from section 2.6.1, meaning that a minimum size is not strictly enforced like e.g. by means of a constraint. However, features that fall below a certain minimum size vanish in the eroded field, thus they do not contribute to the part's stiffness, but they are clearly visible in the dilated field, which is used for the evaluation of the part's volume. Hence, the gained stiffness in relation to the cost in material volume becomes negligible, such that solid features below a certain minimum size are strongly penalized and therefore unfavorable.

The minimum solid feature size dependent on the filter radius r and the projection threshold values can be determined according to a curve given in [31], with the correction given in [86]. An analytic term is not given in the original paper, however, it is found in [87]. Assuming a threshold of $\eta_0 = 0.5$ for the nominal design, the minimum width b of a solid feature can be estimated from the threshold η_e of the eroded design and the filter radius r:

$$\frac{b}{r} = \begin{cases} 2\sqrt{\eta_e - 0.5}, & \eta_e \in [0.5, 0.75] \\ 2 - 2\sqrt{1 - \eta_e}, & \eta_e \in [0.75, 1] \end{cases} \tag{2.23}$$

However, it has to be noted, that the curve in [31] and equation (2.23) are derived considering a 1D density field. For the 2D and 3D case, different relations apply[6], therefore above equation has to be seen more as a loose approximation and not an exact measure. After all, the "robust" approach still cannot hinder smaller features to evolve, as shown e.g. in [79].

Minimum size control is an important topic for optimizations involving compliant mechanisms (see e.g. [88]), where mobility of a structure is obtained by allowing flexibility in some regions, while other regions should remain stiff. Without minimum solid feature size control, unphysical one-node hinges appear due to FE discretization errors.

However, compliant mechanisms are not studied in this work. The "robust" approach is rather used as a tool to obtain designs with a very simple topology, to serve as a reference. Another aspect in the scope of this work is to show, that these designs, despite being optimized to be robust towards deviations by uniform manufacturing errors, are not a good choice when it comes to larger, localized structural deviations imposed by partial damage.

2.7 Example designs for plain and modified topology optimizations

In this section, the previously discussed methods for TO are applied to two widely studied academic examples, namely the 2D cantilever beam and the 2D L-bracket problem. For both examples, the task is to minimize the compliance under a global volume constraint. The results from a plain three field TO serve as a reference. Additionally, optimizations with modified tasks, including the "robust" approach, maximum member size control (cf. section 2.6.3), and redundancy enforcing constraints (cf. section 2.6.2) are performed. Also, a combination of the robust and redundancy enforcing approach is considered. The numerical values of the final compliance for the different optimizations are given and a discussion follows in section 2.7.3.

The filter radius r is set to 4 in all examples and a global volume constraint is used, limiting the volume fraction to $k_g = 0.4$. Table C.1 in Appendix C.6 lists the remaining parameters that are in common.

[6] When seeing the filtering step from equation (2.2) as a convolution integral over the DV field "function" and the cone-shaped filter function from (2.2b), it is obvious, that a convolution with a 2D or 3D cone yields different results compared to a 1D cone (or hat), even if the DV field only has varying values along one dimension.

For the robust approach, threshold values of $\{0.3, 0.5, 0.7\}$ are used for the dilated, nominal and eroded designs (cf. section 2.6.1). In the figures, only the nominal design is depicted and also the given compliances refer to the nominal design.

The maximum member size designs use local volume constraints, limiting the local volume fraction to $k_l = 0.95$ in circular zones with a radius of $R = 8$, requiring all features to have a width smaller than 16.

Redundant designs are also obtained by local volume constraints, but with a local volume fraction of $k_l = 0.5$ within radii of $R = 16$.

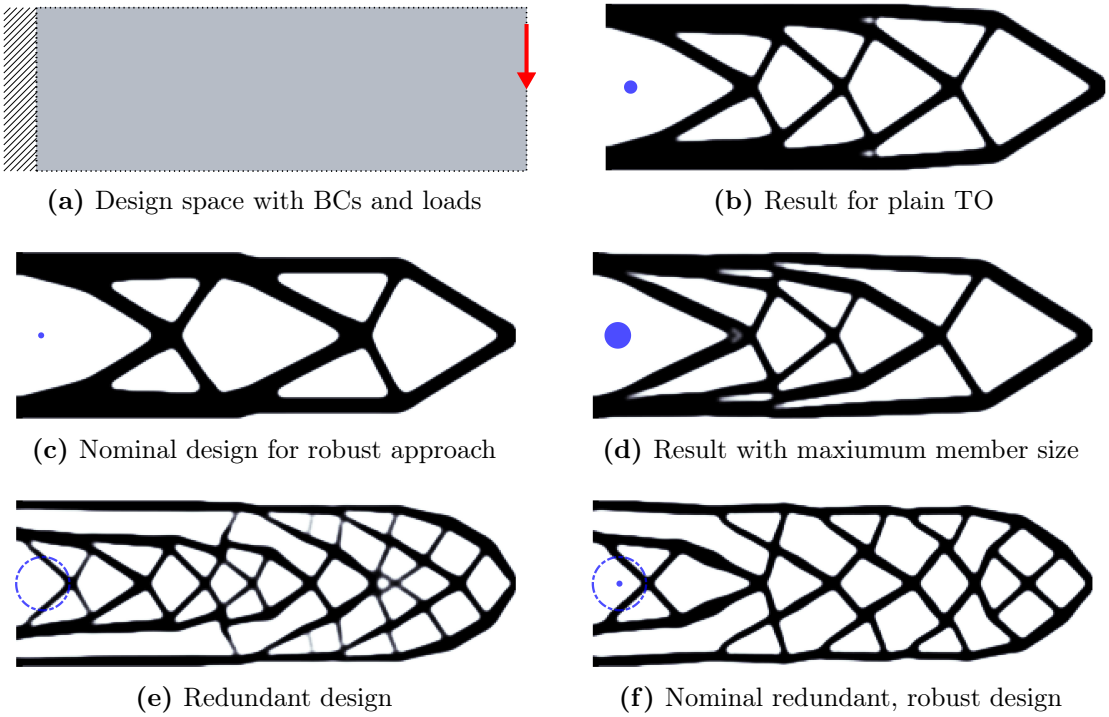
For all examples involving local volume constraints, these are aggregated into a joint constraint using the newly derived aggregation function A_{uni} with a medium valued aggregation parameter of $p = 16$, as explained in the later section 3.5.2.

Blue circles indicate characteristic sizes in the following figures. For the plain TO designs, they refer to the filter size. For TO with the robust approach and for the maximum member size designs, they indicate the corresponding minimum or maximum solid feature size, respectively. Finally, for the redundant designs, dashed circles show the size of the circular test region.

2.7.1 2D cantilever beam example

As the first academic example for compliance minimization under a global volume constraint, a cantilever beam with an aspect ratio of 3×1, discretized by 300×100 finite elements, is studied. The design space is pinned on the left side and a transverse load is acting on the middle node of the right side. The design space with its boundary conditions (BCs) and loads is shown in figure 2.8a.

The individual optimized designs are shown in figures 2.8b – 2.8f. The compliance values c and the actual global volume fractions ω are summarized in table 2.1.



(a) Design space with BCs and loads

(b) Result for plain TO

(c) Nominal design for robust approach

(d) Result with maxiumum member size

(e) Redundant design

(f) Nominal redundant, robust design

Figure 2.8 – Numerical model and results for the cantilever beam example.

Table 2.1 – Compliance values for the cantilever beam example.

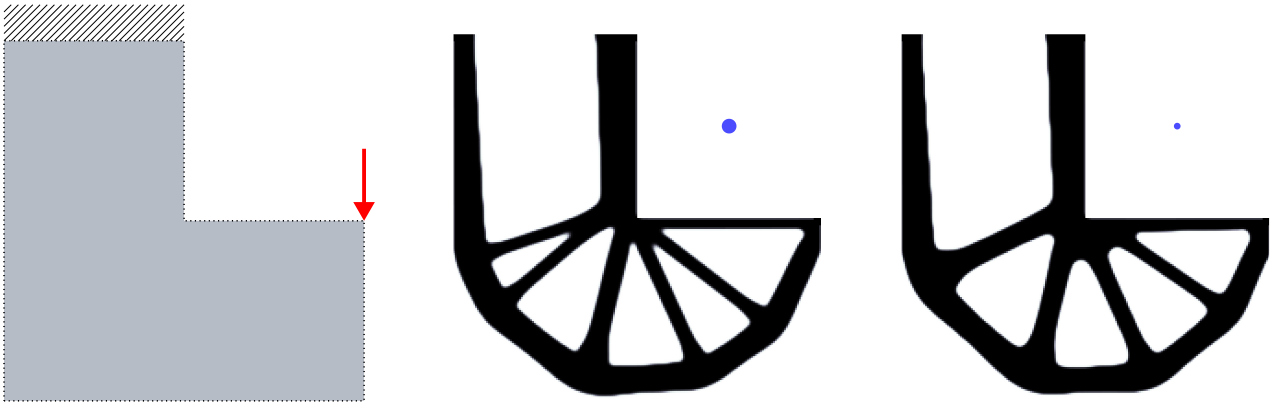
Design	c	ω
Plain TO	210.19	40.00%
Robust approach	213.68	40.02%
Maximum member size	224.47	40.00%
Redundant	328.31	37.17%
Redundant, robust	369.24	35.19%

2.7.2 2D L-bracket example

The second academic example is an L-shaped bracket. The design space, BCs, and loads are shown in figure 2.9a. For the FE analyses, the design space is discretized by 200 finite elements over the total width and height, respectively. The top right quadrant with an outer dimension of 100×100 elements is left out. The total number of elements matches the one for the cantilever example from section 2.7.1.

The L-bracket is an interesting example since asymmetrical designs are obtained and the sharp inner corner is a critical region, where a stress singularity is introduced.

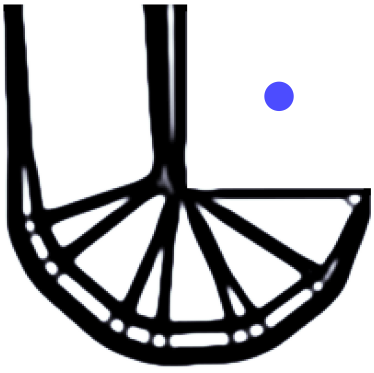
The optimized designs are shown in figures 2.9b – 2.9f. Table 2.2 lists the compliance values c and the global volume fractions ω of the individual designs.



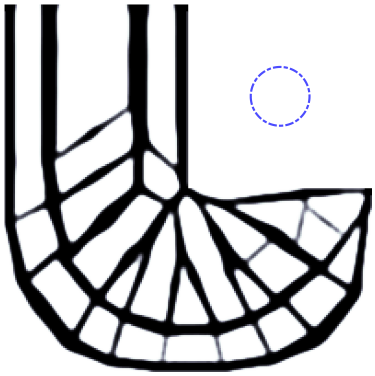
(a) Design space with BCs and loads

(b) Result for plain TO

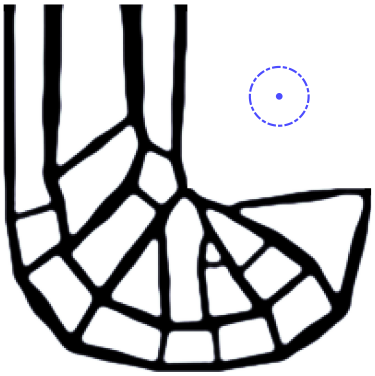
(c) Nominal design for robust approach



(d) Result with maximum member size



(e) Redundant design



(f) Nominal redundant, robust design

Figure 2.9 – Numerical model and results for the L-bracket example.

Table 2.2 – Compliance values for the L-bracket example.

Design	c	ω
Plain TO	100.29	40.00%
Robust approach	100.97	40.02%
Maximum member size	106.44	40.00%
Redundant	152.37	36.26%
Redundant, robust	162.72	35.15%

2.7.3 Discussion

Figures 2.8 and 2.9 show that a wide variety of designs can be obtained by using different formulations of TO tasks. The numerical values in tables 2.1 and 2.2 reveal, that for a plain TO, designs with the lowest compliance are obtained, since the optimization task is the least restricted one. The more restrictive additionally posed constraints are, the worse the compliance of the design becomes.

The designs obtained by the robust approach from figures 2.8c and 2.9c have the simplest topologies since implicitly the surface area is penalized in this approach (see section 2.6.1). The aspect, that also solid features below a minimum size are suppressed is not relevant in this case, since all features of both designs are well above the minimum size. Overall, the compliance is slightly higher than for the plain TO examples. However, the maximum allowed volume for the shown designs is actually slightly exceeded, since the volume constraints are assigned adaptively to the eroded design and not to the nominal design directly, as mentioned in section 2.6.1. Therefore, slight deviations in volume are possible.

The results obtained with a limit on the maximum solid feature size from figures 2.8d and 2.9d have explicit restrictions by local volume constraints additionally to the global volume constraint. Therefore, the feasible domain in the solution space is smaller and a notable increase of the compliance in relation to the reference designs is observed. Thick beams are clearly split into parallel, thinner beams, which are however very close to each other and no fully redundant design is obtained.

In contrast to that, the local volume constraints with larger radii and lower volume fractions employed for the designs in figure 2.8e and 2.9e additionally enforce some minimum separation between the beams, such that redundant, "porous" designs are obtained (cf. section 2.6.2). The restrictions imposed by the local volume constraints dominate the optimizations, such that the global volume constraints are inactive for both designs and slightly less than 40% relative volume is achieved. As a side effect, some very thin members can be seen. The compliance is clearly worse for the redundant designs, compared to the other designs.

Combining the competing goals of enforcing redundant load paths and at the same time using the robust approach to minimize the surface area, the designs in figures 2.8f and 2.9f are obtained. Solid features below the indicated minimum size are suppressed and results with overall fewer features, less global volume, and even higher compliance are obtained. Also, the visually perceived degree of porosity is reduced, compared to the results obtained without the robust approach, which used the same parameters for the local volume constraints. This could be countered e.g. by using smaller radii for the local volume constraints.

For this work, especially the redundant designs are of interest. Despite having the highest compliance of all shown results when considering the intact structures, these designs have the potential to offer superior robustness to partial failure (see [44, 58]). This aspect is thoroughly discussed in sections 4 and 5, which deal with optimizations towards fail-safe designs.

3 Aggregation strategies for objectives and constraints

In general, a smooth aggregation function A can be used to approximate the maximum over a vector of terms $\boldsymbol{\tau}$ in a differentiable form:

$$A(\boldsymbol{\tau}) \approx \max_i \tau_i \tag{3.1}$$

This allows reformulating equations involving the max-operator into alternative, approximate equations based on aggregation functions. These can then be used directly for gradient-based optimization, since A is a differentiable function, unlike the original max-operator.

Using aggregation functions, the accuracy of the approximation of the real maximum value can be improved by increasing an aggregation parameter p (see e.g. figure 3.1 in section 3.3.1). However, choosing too high values may compromise the convergence behavior of the whole optimization in which the aggregation function is used [85]. Hence, there is always a trade-off between accuracy and numerical stability involved, when using aggregation functions.

The gradient of A with respect to the DVs can be seen as a weighted linear combination of the gradients of the n_t individual input terms τ_j:

$$\frac{dA(\boldsymbol{\tau})}{d\varrho_i} = \sum_{j=1}^{n_t} \underbrace{\frac{\partial A}{\partial \tau_j}}_{w_j} \frac{\partial \tau_j}{\partial \varrho_i} = \sum_{j=1}^{n_t} w_j(\boldsymbol{\tau}) \frac{\partial \tau_j}{\partial \varrho_i} \tag{3.2}$$

The weights w_j themselves are dependent on the distribution of the terms in $\boldsymbol{\tau}$ and the choice of the aggregation parameter p. Generally speaking, the weight is highest for the input term with the maximum value and then quickly decreases for terms with increasingly lower values.

Two general use cases for aggregation functions with relevance for this work are discussed in section 3.1: Aggregation of objectives and aggregation of constraints. Commonly used aggregation functions are presented in section 3.2. Their important properties are highlighted in section 3.3.

For some applications, very high values for the aggregation parameter p are reported in the literature to achieve the desired accuracy (see e.g. [85], where values up to $p = 300$ are used). In this work, this is seen as an indication, that the chosen aggregation function is not well suited for the considered distribution of input values. Therefore, in section 3.4, a method is presented to derive new, case specific aggregation functions if information about the input data distribution is available or can be at least estimated. With these tailored aggregation functions, significantly lower values of the aggregation parameter are sufficient to achieve the desired accuracy, while the overall numerical stability of the optimization is improved.

Finally, in section 3.5, the explicit choice of aggregation functions and their parameters for specific applications that are in the focus of this work are explained.

3.1 Relevant use cases

Two general use cases for aggregation functions are relevant for this work and will be discussed in this section:

1. Aggregation of many objective function values into a single value resembling the maximum (worst-case) objective value
2. Aggregation of a large number of similar constraint functions into a single joint constraint

3.1.1 Aggregation of objective function values

Aggregation of objective values is used to cope with min-max optimization problems. In such problems, the objective function term[7] τ_i with the maximum (or worst) value should be minimized by altering the design vector $\boldsymbol{\varrho}$. To obtain a differentiable function that can be used for gradient-based optimization, the max-operator is replaced by a differentiable approximation through a suitable aggregation function A:

$$\min_{\boldsymbol{\varrho}} \max_{i} \tau_i(\boldsymbol{\varrho}) \approx \min_{\boldsymbol{\varrho}} A\left(\boldsymbol{\tau}(\boldsymbol{\varrho})\right) \tag{3.3}$$

For such worst-case optimization tasks, using a smoothed approximation of the real maximum operator has the additional advantage, that terms with values close to the currently worst term still have an influence on the aggregated function value and its gradients and therefore stay "visible" for the optimization algorithm. In contrast to that, using an active-set strategy and considering solely the worst term in each iteration would minimize this term at the expense of other terms not considered, which would jeopardize convergence, if this results in a configuration with constant switching between the same two terms.

The accuracy of the aggregation function A in equation (3.3) is usually of minor importance for this use case. As long as A is strongly biased towards the real maximum term, minimizing the aggregated value will also minimize the worst-case term, even if A adds some offset to the real maximum. The focus of the aggregation lies in this case rather on achieving desired properties regarding the combination of the gradients of the individual terms (cf. equation (3.2)) than on the accuracy of the aggregated value.

3.1.2 Aggregation of constraint functions terms

Aggregation of constraint functions can be used in cases, where n_t constraints h_i exist, that limit different design response terms τ_i, dependent on the DVs $\boldsymbol{\varrho}$, to the same constant threshold value t:

$$h_i(\boldsymbol{\varrho}) = \tau_i(\boldsymbol{\varrho}) - t \leq 0, \quad i = 1, \ldots, n_t \tag{3.4}$$

Whether a solution satisfies these n_t constraints is determined by the terms τ_i with the highest values, since these appear in the constraints that are violated the most or are the nearest to being violated, respectively. In using a suitable aggregation function A, the n_t constraints can be condensed in a single constraint h_{agg}:

$$h_{agg}(\boldsymbol{\varrho}) = A\left(\boldsymbol{\tau}(\boldsymbol{\varrho})\right) - t \leq 0 \tag{3.5a}$$

$$\approx \max_{i} \tau_i(\boldsymbol{\varrho}) - t \leq 0 \tag{3.5b}$$

Constraint aggregation is therefore a way to greatly reduce the size of the dual problem of the optimization task (and thus the computational time), if a large number of similar constraints with the same threshold value t are present.

[7] In the context of objectives, "term" is used as a generic name and may either refer to the individual entries of a vector, if the studied objective function returns a vector-valued result or to individual scalar objective function values, if an objective function is evaluated for several distinct configurations, like e.g. for the robust approach (cf. eq. (2.19)) or when considering damage (cf. eqs. (4.5) and (5.9)).

Depending on the use case, the threshold t may be a limit that must not be exceeded in any case. Then, an aggregation function A with a conservative mapping is needed and the focus lies on the precision of the approximation. For other use cases, some degree of violation may be acceptable. The desired behavior can be tuned by a proper choice of the function for A and of its aggregation parameter p.

3.2 Commonly used aggregation functions

Commonly used aggregation functions in the context of TO are:

1. P-norm aggregation function (A_{pn})
2. P-mean aggregation function (A_{pm})
3. Kreisselmeier-Steinhauser aggregation function (A_{KS})

All three functions share the property, that input values are mapped via a progressive, nonlinear function, summed up, optionally scaled by a factor, and then mapped back by the inverse of the nonlinear function. The progression of the nonlinear mapping function is dependent on an aggregation parameter p.

The aggregation parameter on the one hand controls, how well the aggregation function approximates the real maximum of all input terms $\boldsymbol{\tau}$. On the other hand, it also influences the way how the gradients of the input terms are combined into a joint gradient.

The definitions of the above-mentioned three aggregation functions are given in the following subsections, together with mathematical expressions for the gradient weights from equation (3.2), which are derived in Appendix D.6.

3.2.1 P-norm aggregation function

The p-norm is a mathematical norm and is defined as:

$$\|\boldsymbol{\tau}\|_p := \left(\sum_i |\tau_i|^p \right)^{\frac{1}{p}} \tag{3.6}$$

For increasing values of $p \to \infty$ the p-norm approaches the maximum of the absolute values of all τ_i.

Assuming, that the input vector $\boldsymbol{\tau}$ is composed of n_t non-negative real numbers, the absolute value operator may be omitted and the p-norm aggregation function A_{pn} is obtained:

$$A_{pn}(\boldsymbol{\tau}) = \left(\sum_{i=1}^{n_t} \tau_i^p \right)^{\frac{1}{p}}, \quad \tau_i \geq 0 \tag{3.7}$$

The weights for the gradient (cf. equation (3.2) and Appendix D.6) are:

$$w_j = \left(\sum_{k=1}^{n_t} \tau_k^p \right)^{\frac{1}{p}-1} \tau_j^{p-1} \tag{3.8}$$

3.2.2 P-mean aggregation function

The p-mean aggregation function A_{pm} differs from A_{pn} in the fact, that a scaling factor of $1/n_t$ is applied prior to the reverse mapping:

$$A_{pm}(\boldsymbol{\tau}) = \left(\frac{1}{n_t} \sum_{i=1}^{n_t} \tau_i^p \right)^{\frac{1}{p}}, \quad \tau_i \geq 0 \tag{3.9}$$

The weights for the gradient (cf. equation (3.2) and Appendix D.6) are then:

$$w_j = \left(\frac{1}{n_t} \right)^{\frac{1}{p}} \left(\sum_{k=1}^{n_t} \tau_k^p \right)^{\frac{1}{p}-1} \tau_j^{p-1} \tag{3.10}$$

3.2.3 Kreisselmeier-Steinhauser aggregation function

The Kreisselmeier-Steinhauser (KS) function [89,90], which is also known as exponential penalty function in the context of nonlinear optimization [91], is again strongly linked to the p-norm function. The KS aggregation function A_{KS} involves a second mapping step, in which the input values are at first mapped by the exponential function prior to taking the p-norm of the values. Afterward, a reverse mapping with the natural logarithm is performed [92]:

$$A_{KS}(\boldsymbol{\tau}) = \log\left(A_{pn}(\mathrm{e}^{\boldsymbol{\tau}})\right) \tag{3.11a}$$

$$= \log\left(\left(\sum_{i=1}^{n_t} (\mathrm{e}^{\tau_i})^p \right)^{\frac{1}{p}} \right) \tag{3.11b}$$

$$= \frac{1}{p} \log\left(\sum_{i=1}^{n_t} \mathrm{e}^{p\tau_i} \right), \quad \tau_i \in \mathbb{R} \tag{3.11c}$$

The aggregation parameter p is referred to as γ in the original references [89,90]. The biggest advantage of the additional mapping with the exponential function is, that A_{KS} also properly handles input values with negative or alternating signs.

The weights for the gradient (cf. equation (3.2) and Appendix D.6) of A_{KS} are:

$$w_j = \frac{\mathrm{e}^{p\tau_j}}{\sum_{k=1}^{n_t} \mathrm{e}^{p\tau_k}} \tag{3.12}$$

3.3 Properties of aggregation functions

In this section, relevant properties of the three aggregation functions from section 3.2 are summarized. An extended discussion of properties can be found in [93] and specifically for the KS function in [94].

3.3.1 Estimation properties

The p-norm and KS aggregation functions (A_{pn} and A_{KS}) are differentiable, conservative overestimators of the maximum-operator, i.e. they yield a result that is greater or equal to the real maximum value, considering non-negative input values (A_{KS} also works for negative numbers) and $p \geq 1$:

$$A_{pn}(\boldsymbol{\tau}) \geq \max_i \tau_i, \quad \tau_i \geq 0 \tag{3.13a}$$

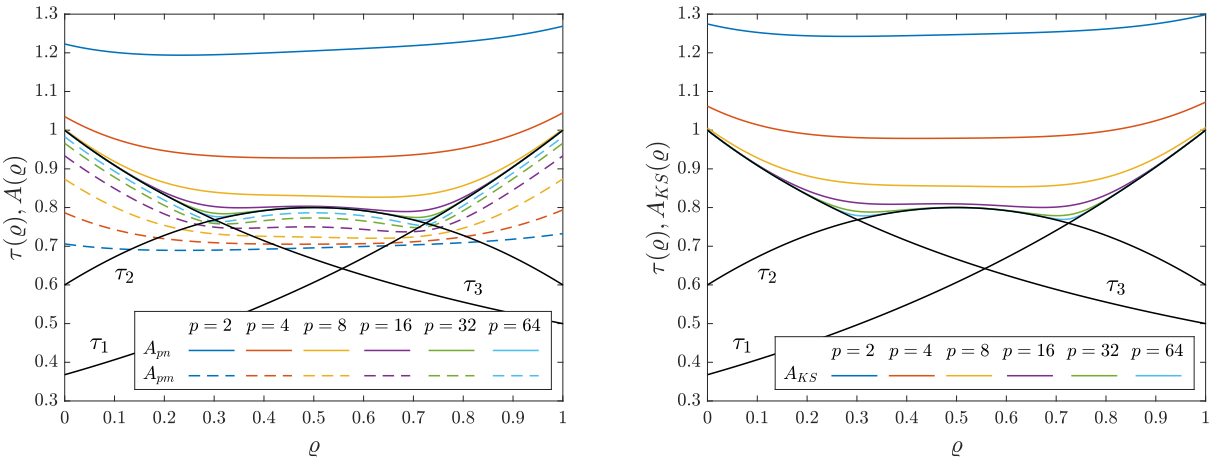
$$A_{KS}(\boldsymbol{\tau}) \geq \max_i \tau_i, \quad \tau_i \in \mathbb{R} \tag{3.13b}$$

For A_{pn} and A_{KS}, any additional nonzero entry appended to the input vector $\boldsymbol{\tau}$ increases the resulting value (cf. equations (3.7) and (3.11c)), therefore these aggregation functions are conservative.

Contrary to that, A_{pm} underestimates the maximum value. Equation (3.9) shows, that, with the expression in the large bracket, the mean value of the progressively mapped input values is considered in the reverse mapping step. Therefore, any input value lower than the maximum over $\boldsymbol{\tau}$ drags the resulting value down, and the following holds:

$$A_{pm}(\boldsymbol{\tau}) \leq \max_i \tau_i, \quad \tau_i \geq 0 \tag{3.14}$$

Figure 3.1 shows the effect of increasing the aggregation parameter p on the estimation characteristics of A_{pn}, A_{pm}, and A_{KS} for an exemplary aggregation of three terms τ_i, dependent on a single DV ϱ.



(a) Aggregation with p-norm and p-mean

(b) Aggregation with KS

Figure 3.1 – Example for the aggregation of three terms τ_1, τ_2, and τ_3 for varying p values.

The accuracy of approximating the maximum over the input terms increases for higher p values for all three aggregation functions, as shown in figure 3.1. This is due to the progressive nature of processing the input values by taking them to a power of $p \gg 1$ (cf. equations (3.7), (3.9), and (3.11b)). This amplifies the relative difference of the largest input values to the remaining values, thus biasing the result increasingly towards the large values.

3.3.2 Influence of the scale of the input terms

The aggregation functions A_{pn} and A_{pm} are linear with respect to the scale of the input terms, thus, a scaling factor $s > 0$ can be pulled out:

$$A_{pn}(s\boldsymbol{\tau}) = sA_{pn}(\boldsymbol{\tau}) \tag{3.15a}$$
$$A_{pm}(s\boldsymbol{\tau}) = sA_{pm}(\boldsymbol{\tau}) \tag{3.15b}$$

Therefore, normalized terms in the range of $[0, 1]$ can be used with these functions without affecting the accuracy. Using normalized values circumvents the problem of a numerical overrun for large p and a large scale of the input values.

The KS function A_{KS} is not linear with respect to the scale of the input values, a scaling factor cannot be pulled out, since it corresponds to an altered exponent of sp for the exponential function in equation (3.11c). However, the obtained result for a scaled input vector still converges to the scaled maximum value for an increase of the aggregation parameter p:

$$A_{KS}(s\boldsymbol{\tau}) \neq sA_{KS}(\boldsymbol{\tau}) \tag{3.16a}$$
$$A_{KS}(s\boldsymbol{\tau}) \approx s\max(\boldsymbol{\tau}) \tag{3.16b}$$

For the KS function, a change of the scale of the input values has an effect on the accuracy of the approximation of the maximum of $\boldsymbol{\tau}$. Therefore, the choice of the aggregation parameter p for the KS function is often tuned depending on the expected order of the input values (see e.g. [35]).

3.3.3 Influence of the number of input terms

In contrast to A_{pn} and A_{KS}, the averaging with the factor $1/n_t$ in A_{pm} (cf. equation (3.9)) makes the results of latter function independent of the number of input terms, as long as their values follow the same distribution. This becomes easily evident, when comparing the results for an input vector $\boldsymbol{\tau}$ and another vector $\boldsymbol{\mathcal{T}}$ with duplicated values (and assuming $p \gg 1$):

$$\boldsymbol{\tau} \in \mathbb{R}^+, \quad \boldsymbol{\mathcal{T}} = \begin{bmatrix} \boldsymbol{\tau} \\ \boldsymbol{\tau} \end{bmatrix} \tag{3.17a}$$
$$A_{pn}(\boldsymbol{\mathcal{T}}) = 2^{\frac{1}{p}} A_{pn}(\boldsymbol{\tau}) > A_{pn}(\boldsymbol{\tau}) \tag{3.17b}$$
$$A_{KS}(\boldsymbol{\mathcal{T}}) = A_{KS}(\boldsymbol{\tau}) + \frac{\log(2)}{p} > A_{KS}(\boldsymbol{\tau}) \tag{3.17c}$$
$$A_{pm}(\boldsymbol{\mathcal{T}}) = A_{pm}(\boldsymbol{\tau}) \tag{3.17d}$$

Since A_{pn} and A_{KS} are already overestimating the real maximum value (see section 3.3.1), the precision of these aggregation functions further decreases for large quantities of input terms. This can be countered by an increase of the aggregation parameter p at the price of a higher degree of nonlinearity of the aggregation function, possibly leading to convergence issues in the optimization.

3.4 Alternative aggregation functions

When applying aggregation in practice, it is observed, that the p-norm aggregation function is overestimating the real maximum value of the input terms and the p-mean function is underestimating it (see e.g. figure 3.1a). Hence, it is assumed that, as a midway, an alternative aggregation function can be found that yields a more accurate approximation of the maximum.

In the literature, a wide variety of values for the aggregation parameter p is reported, implying, that p was successively increased up to high values, in cases, where the desired accuracy was not achieved. In [85] for instance, a value of $p = 100$ is reported for one example and $p = 300$ for another one[8].

This work however follows another approach: Instead of making unsuitable aggregation function work by choosing high p values, rather slightly modified aggregation functions are proposed that increase the accuracy of the aggregation for any p value in taking into account additional information about the input data distribution. Therefore, at first, in section 3.4.1 it is analyzed, for which distributions the known p-norm and p-mean aggregation function yield their respective highest accuracy, i.e. zero error. Then, a general "p-like" aggregation function is presented in section 3.4.2, offering an additional tuning factor. In section 3.4.3, the value of this factor for the case of uniformly distributed input terms is presented together with the corresponding aggregation function.

3.4.1 Ideal data distributions for p-norm and p-mean aggregation functions

In this section, the inherent properties of the known aggregation functions A_{pn} (p-norm) and A_{pm} (p-mean) are analyzed, which are valid for any choice of $p \geq 1$. For the following calculations, let $\bar{\tau}$ denote the real maximum value of all n_t non-negative input terms τ_i:

$$\bar{\tau} := \max_i \tau_i, \quad \tau_i \geq 0, \quad i = 1, \ldots, n_t \tag{3.18}$$

The KS aggregation function A_{KS} is seen as a special case of the p-norm aggregation function (cf. section 3.2.3) and is thus not treated separately in the following discussion.

P-norm aggregation function The p-norm aggregation function A_{pn} from equation (3.7) yields the exact maximum (i.e. the highest accuracy is achieved), if the term inside the large bracket is equal to $\bar{\tau}^p$:

$$A_{pn}(\boldsymbol{\tau}) = \left(\sum_{i=1}^{n_t} \tau_i^p \right)^{\frac{1}{p}} \stackrel{!}{=} (\bar{\tau}^p)^{\frac{1}{p}} = \bar{\tau} \tag{3.19}$$

It is obvious, that equation (3.19) is only fulfilled, if the input vector $\boldsymbol{\tau}$ is composed of all zeros, except for one single positive value, which in this case is automatically the maximum value:

$$\boldsymbol{\tau} = \left[0, \cdots, 0, \bar{\tau}\right]^T \tag{3.20}$$

[8] With an exponent as high as 300, a numerical overflow for the standard double-precision data type used in MATLAB occurs already for a basis with a value of approximately 10.6541, yielding *inf* as result.

In this case, independently of p, the exact maximum value is retrieved. Any (positive) deviation from the zero values of the vector in (3.20) increases the value of the sum from equation (3.19) and therefore leads to an overestimation of the real maximum. The largest error for the p-norm aggregation function A_{pn} is obtained, if all values of the input vector $\boldsymbol{\tau}$ are of the same, maximum magnitude $\bar{\tau}$ (see the vector in (3.21)). In this case the overestimation by A_{pn} involves a factor of ${n_t}^{\frac{1}{p}}$.

P-mean aggregation function For the p-mean aggregation function A_{pm} from equation (3.9), the opposite holds, when compared to the p-norm aggregation function: The ideal input vector $\bar{\tau}$ has only entries with the maximum value $\bar{\tau}$:

$$\boldsymbol{\tau} = \left[\bar{\tau}, \cdots, \bar{\tau}\right]^T \tag{3.21}$$

In this case, A_{pm} yields the exact maximum value, since the expression in the brackets takes on the desired value of $\bar{\tau}^p$, independently of the choice of p, and following equation is fulfilled:

$$A_{pm}(\boldsymbol{\tau}) = \left(\frac{1}{n_t}\sum_{i=1}^{n_t}\tau_i^p\right)^{\frac{1}{p}} \stackrel{!}{=} (\bar{\tau}^p)^{\frac{1}{p}} = \bar{\tau} \tag{3.22}$$

If any value τ_i has a lower value than the ideal value $\bar{\tau}$, the expression in the brackets of equation (3.22) is decreased, therefore, the real maximum will be underestimated by using A_{pm}. The worst case is a vector with only one large entry like the vector from (3.20). The factor of underestimation is then ${n_t}^{-\frac{1}{p}}$.

Conclusion Both studied aggregation functions achieve their highest accuracy in approximating the max-operator for the most extreme skew value distributions: For the case, that all values are zero, except for a single value, or the case that all values are equal, respectively.

In practical application, the input value distribution will most often strongly deviate from these "ideal" input vectors, since also many values between zero and the maximum value are present, which gives rise to approximation errors. In these cases, p has to be increased to make use of the convergence characteristics discussed in section 3.3.1 to achieve the desired accuracy. This work however proposes to reconsider the aggregation function definitions and to derive alternative functions, that are better suited for input distributions encountered in practice.

3.4.2 A general p-like aggregation function

Considering, that the p-norm (A_{pn}) and p-mean (A_{pm}) aggregation functions only differ by a factor prior to the sum operator, a general "p-like" aggregation function with a correction factor α, which is yet to be determined, is defined:

$$A_g(\boldsymbol{\tau}) = \left(\alpha\sum_{i=1}^{n_t}\tau_i^p\right)^{\frac{1}{p}}, \quad \tau_i \geq 0, \quad \alpha \in \left[\frac{1}{n_t}, 1\right] \tag{3.23}$$

The functions A_{pn} and A_{pm} can thus be seen as special cases of the general aggregation function A_g, when α is chosen to be 1 or $1/n_t$, respectively. In limiting α to the above-given

interval, the p-norm and p-mean aggregation functions form an envelope to A_g. The factor α now acts as an additional tuning factor and can be determined, if the distribution of the values in $\boldsymbol{\tau}$ is known or can at least be estimated.

The derivation of α values for different input value distribution is described in detail in Appendix A.2. For this work, only one additional case is relevant, namely the aggregation of uniformly distributed values, which is discussed in the following section 3.4.3. In Appendix A.2, also α values for moderately skew deviations from uniform distributions are derived, which approach the ideal distributions for A_{pn} and A_{pm} shown above. However, the corresponding aggregation schemes are not used in this work.

The proposed procedure of deriving a constant correction factor α in advance, based on an assumption for the data distribution, differs substantially from other approaches reported in the literature, where an aggregation function correction factor is fitted at the end of each iteration to minimize the difference of the aggregated and the true maximum value (see e.g. [95]). In the latter case, the constraint function is exchanged unsteadily, which may jeopardize convergence.

3.4.3 Aggregation function for uniformly distributed input values

Aggregation functions ideal for input terms with very skew value distributions are already covered by the p-norm and p-mean functions, as discussed in section 3.4.1. However, for uniform or close to uniform value distributions, neither of the two just mentioned aggregation functions is particularly well suited. Therefore, as a third choice, an aggregation function especially tailored for uniform distributions as a "midway" is given in this section, to allow a broader application of p-like aggregation functions.

Assuming, that the input terms in $\boldsymbol{\tau}$ are uniformly distributed in an interval $[0, \bar{\tau}]$, the specific value of the correction factor α from equation (3.23) to recover the maximum value of $\boldsymbol{\tau}$ is derived in Appendix A.2. The obtained value is:

$$\alpha = \frac{p+1}{n_t} \tag{3.24}$$

Inserting this value into the general aggregation function from equation (3.23) yields the "p-uniform" aggregation function A_{uni} specifically suited for uniformly distributed values:

$$A_{uni}(\boldsymbol{\tau}) = \left(\frac{p+1}{n_t} \sum_{i=1}^{n_t} \tau_i^p \right)^{\frac{1}{p}}, \quad \tau_i \geq 0 \tag{3.25}$$

Considering the aggregation parameter p to be in the range of $1 \leq p \leq n_t - 1$, the factor α from equation (3.24) meets the requirement of being in the range given in equation (3.23). In this case, the A_{pn} and A_{pm} aggregation functions act as upper and lower bounds on A_{uni}.

The estimation properties of A_{uni} now depend on the distribution of the input data. If the input data exactly follows a uniform distribution, the proposed aggregation function exactly retrieves the real maximum value, independently of the specific choice for the parameter p (for details see the derivation in Appendix A.2), at least for a large number of terms n_t. In cases, where deviations from the ideal input distribution are present, A_{uni} inherits the properties from A_{pn} and A_{pm}, namely, that it still converges towards the real maximum for increasingly larger p values. However, specific estimation characteristics (i.e. strict overestimation or strict underestimation) cannot be guaranteed for A_{uni}, it is only known, that the resulting aggregated value lies between the ones obtained by A_{pn} and A_{pm}.

A numerical test of the aggregation function A_{uni} is given in Appendix A.3.

3.5 Aggregation schemes applied in this work

The choice of a specific aggregation function and a value for its aggregation parameter p may be critical for certain applications. Choosing a too low value for p may lead to large errors in the approximation, rendering e.g. an aggregated constraint unusable. However, also a too large p value has a negative effect on the optimization since then the aggregated function corresponds more and more to an active-set strategy involving a switching condition, in which only the currently largest function term is visible to the optimizer and thus drives the whole optimization. Therefore, very large p values should also be avoided.

In the literature, a wide range of p values is reported, sometimes changing from one considered example to the other, indicating, that only a narrow interval of values yield satisfactory results. In sections 3.5.1 and 3.5.2, the aggregation of worst-case objective values and the aggregation of local volume constraints, respectively, are discussed as relevant main use cases for aggregation strategies in this work and the used aggregation functions and their parameters are given transparently.

A use case beyond the scope of this work is the aggregation of stress constraints. Typically, multiple constraints per element are considered in such stress-constrained problems. However, to achieve a conservative approximation demands the use of aggregation functions whose performance diminishes with the number of constraints (p-norm or KS aggregation function). Therefore, large-scale stress-constrained problems cause an obstacle for aggregation strategies such that currently augmented Lagrange (AL) methods move into the focus of research (see e.g. [96, 97]). Hence, stress constraint aggregation is not covered in this section.

3.5.1 Aggregation of worst-case objective values

Worst-case optimization problems appear e.g. in the context of the robust approach for TO (see section 2.6.1) and in fail-safe optimizations considering damage (see sections 4.1.4, 5.2.2, and 6.7). In these problems, minimizing the objective function Φ_i with the maximum (worst-case) value leads to a min-max problem taking into account all n_t considered scenarios (cf. equation (3.3)):

$$\min_{\boldsymbol{\varrho}} \max_{i=1,\dots,n_t} \Phi_i(\boldsymbol{\varrho}) \tag{3.26}$$

As explained in section 3.1.1, using aggregation is a way to replace the non-differentiable max-operator in the above statement with a differentiable function[9]. A specific aggregation function can be chosen dependent on the expected distribution of the objective values Φ_i.

In the problems studied in this work, the worst-case scenarios yield objective values that are orders of magnitude higher than the values obtained for the majority of the other scenarios. The corresponding distribution of objective values Φ_i is therefore expected to consist of only a small number of extremely high values, while most values are comparably low. Suitable candidates to aggregate such a distribution are therefore the p-norm function (A_{pn} from section 3.2.1) or the related KS aggregation function (A_{KS} from section 3.2.3). Both aggregation functions yield conservative approximations to the max-operator.

For optimization objectives with strictly positive values, the choice of the p-norm or KS aggregation function is of minor importance (see e.g. the studies in [98, 99]), since for both functions

[9] Another approach to tackle min-max problems is to use a slack variable approach, as mentioned in section 2.5. However, for most of the studied problems, this would significantly increase the computational cost of the dual problem in the optimization since the number of terms n_t is large. Therefore, this approach is not followed in this work.

an aggregation parameter p can be found to achieve a similar and, above all, quickly decaying weighting of the individual input terms. This property is analyzed in detail in Appendix A.1.

Nevertheless, the KS function A_{KS} can also handle input values with alternating signs, unlike the p-norm function A_{pn}. Although negative objective values do not appear for compliance optimization problems, they are encountered e.g. in displacement optimization problems involving compliant mechanisms (see e.g. [75, 88]). Therefore, for the sake of generality, the KS function is used for objective function aggregation in the context of worst-case optimization throughout this work.

Using the KS aggregation function A_{KS} to calculate an aggregated objective function Φ_{agg}, the task for a worst-case optimization from statement (3.26) is altered to:

$$\min_{\boldsymbol{\varrho}} \underbrace{\frac{1}{p} \log \left(\sum_{i=1}^{n_t} \mathrm{e}^{p\Phi_i(\boldsymbol{\varrho})} \right)}_{\Phi_{agg}} \tag{3.27}$$

The aggregation parameter p in equation (3.27) is used to control how quickly non-critical scenarios lose influence on the aggregated objective value and, more importantly, on its combined gradient.

If p is chosen sufficiently high, the aggregated objective and its gradient is dominated by the contribution of the worst-case scenario. The influence of non-critical scenarios quickly diminishes, which is a desired property for this use case. Formulae to quantify this decaying behavior in relation to the aggregation parameter p are given in Appendix A.1.

This work adopts the recommendation from [35] to choose a problem-specific value of p for the KS function that is tuned in relation to the expected order of the highest objective values of all Φ_i, such that the exponent in equation (3.27) stays in a numerical range between 1 and 100, preferably close to a value of 50. Consequently, constant values of e.g. $p = 50/10^3$ or $p = 50/10^4$ are used in this work.

3.5.2 Aggregation of local volume constraints

In the context of this work, local volume constraints (see section 2.4.3) are used to enforce redundant structures and control maximum member sizes in TO problems. While for both targets the exact same constraint functions apply, the difference lies in the choice of the threshold value k_l for the maximum local volume fraction and the required strictness of the constraint enforcement in the optimization.

Using local volume constraints, an aggregation strategy is mandatory since usually one constraint is employed per finite element. As a result, in the literature, the p-mean aggregation function (A_{pm} from section 3.2.2) is chosen to aggregate local volume constraints, since its performance does not diminish with an increase of the number of aggregated constraints. However, A_{pm} is not a conservative aggregation function. Consequently, the maximum allowed local volume fraction will be slightly exceeded in some regions of the design.

To what extent this excess is acceptable depends on the use case: If local volume constraints are only used as a means to generate redundant TO designs, the accuracy of the aggregation is of minor importance to achieve the desired effect. Therefore, a medium-valued choice of the aggregation parameter p is sufficient, e.g. $p = 16$ is given in the original paper [44]. In contrast to that, for use in the context of maximum member size control, balancing the strictness of the aggregated local volume constraint is vital since k_l is close to the upper bound (a value of 0.95 is used in this work). Choosing a too low p value means, that also completely solid control

regions are accepted, rendering the aggregated constraint useless, since then arbitrarily thick members are accepted in the design. However, increasing the strictness too much by choosing very high p values amplifies undesired effects such as undercuts at intersections of solid features (cf. [100]) and perforated solid features (see [85]). Finding a suitable p value a priori proves to be difficult, since reported values vary from one example to the other, like e.g. in [85], where values up to $p = 300$ are used.

The main cause for above-mentioned problem is, that the aggregation function A_{pm} is a suboptimal choice when dealing with local volume constraints in this context. As shown in section 3.4.1, A_{pm} is ideal for distributions with mainly equal values. However, when dealing with local volume fractions, a broad spectrum of values is expected, ranging from 0 for purely void regions up to 1 for purely solid regions, including everything in between. None of the existing commonly used aggregation functions from section 3.2 is particularly well suited for such a distribution.

The numerical studies in this work show, that the proposed aggregation function for uniformly distributed values (A_{uni} from section 3.4.3) yields more accurate results at much lower p values compared to A_{pm} when aggregating local volume constraints. While most certainly the distribution of local volume fractions is not exactly uniform, it is way closer to it, than to the extreme skew distributions ideal for A_{pn} or A_{pm} (cf. section 3.4.1). The obtained results further underline, that more exact modeling of the distribution is not necessary for the use cases considered since A_{uni} already performs well using a medium high aggregation parameter of $p = 16$.

Consequently, A_{uni} is used as a more general choice to aggregate local volume constraints throughout this work. Independently of the use case, A_{uni} with the same medium value of $p = 16$ is successfully applied in the context of redundancy enforcement as well as for maximum member size control, which is not possible with A_{pm}. In using a relatively low value for p, the numerical stability of the aggregated constraint is improved, which results in an overall better convergence of the whole optimization. Additionally, A_{uni} inherits the property of A_{pm}, that its performance is not penalized by an increase in the number of input terms.

Assuming, that one local volume constraint per finite element is requested, and with $\boldsymbol{\omega}$ being the vector of all n_e corresponding local volume fractions according to equation (2.16b), the aggregated local volume constraint h_{agg} states:

$$h_{agg} = A_{uni}(\boldsymbol{\omega}) - k_l \leq 0, \qquad \boldsymbol{\omega} = [\omega_1, \ldots, \omega_{n_e}]^T \tag{3.28a}$$

$$= \left(\frac{p+1}{n_e} \sum_{i=1}^{n_e} \omega_i^p \right)^{\frac{1}{p}} - k_l \leq 0 \tag{3.28b}$$

4 Topology optimization methods considering partial damage

This section covers methods to include damage considerations directly inside a TO, which allows optimizing parts with the property that their structural response stays within desired limits, even if a partial failure occurs.

Section 4.1 presents an extension to the standard three field TO approach from section 2, allowing damaged configurations to be considered directly during the optimization.

How the damage scenarios are defined varies depending on the chosen damage model. The approach of Jansen et al. from [35] uses a regular grid of fixed, rectangular geometries, in which the elements are switched to void elements in the case of failure. This method is discussed in section 4.2, where also two strategies for efficiency improvement are presented and assessed.

Nonetheless, methods based on the damage model from [35] couple the number of damage scenarios to the discretization of the FE mesh, which rules out an application for high-resolution large-scale structures in an industrial context. Therefore, a new method by the author, involving adaptively shaped damage geometries covering the load paths of a part (see [58,101]), is presented in section 4.3. This damage model can significantly reduce computational time since only a much smaller number of failure cases needs to be considered that is dependent on the number of structural features of a part. Additionally, the proposed damage model is closer to the initial intention of checking for failure on one load-bearing feature after the other. Most parts of section 4.3 are recapitulated from a preceding publication of the author in [58].

The application of the load-path-based damage model in a TO is shown in section 4.3.5, where a discussion gives rise to a novel optimization method which is then covered in section 5.

4.1 Extending the TO framework by damaged configurations

Figure 4.1 shows, how structural damage for a topology optimization considering failure is modeled in this work.

The solid material of the optimized part is distributed inside the design domain Ω and is described by the projected density field $\bar{\boldsymbol{\varrho}}$ (see section 2.3). Damage is applied to the structure by removing any solid material that lies inside of predefined zones ${}^{-}\Omega_d$, where the index d stands for one of the n_d total damage scenarios considered[10]. This models a complete and instantaneous degradation of the stiffness properties of the material inside the damage zone towards the stiffness properties of the void phase.

The damage zones ${}^{-}\Omega_d$ are removed one at a time. Two different approaches to model these damage zones and the definition of their respective geometry are discussed in this work: Section 4.2 covers the simplified rectangular damage model according to [35]. A newly introduced load-path-based parametric damage model from [58] is presented in section 4.3.

4.1.1 The fourth field of modified densities for TO considering failure

In this work, the resulting material distribution fields obtained by removing the damage zones are referred to as "modified densities" $\hat{\boldsymbol{\varrho}}$.

The modified densities $\hat{\boldsymbol{\varrho}}_d$ resemble the damaged configuration for a damage scenario d and are derived from the ideal, intact projected density field $\bar{\boldsymbol{\varrho}}$ by removing the corresponding material

[10] The upper left "-" symbol in ${}^{-}\Omega_d$ indicates, that these sets designate material zones that are to be removed.

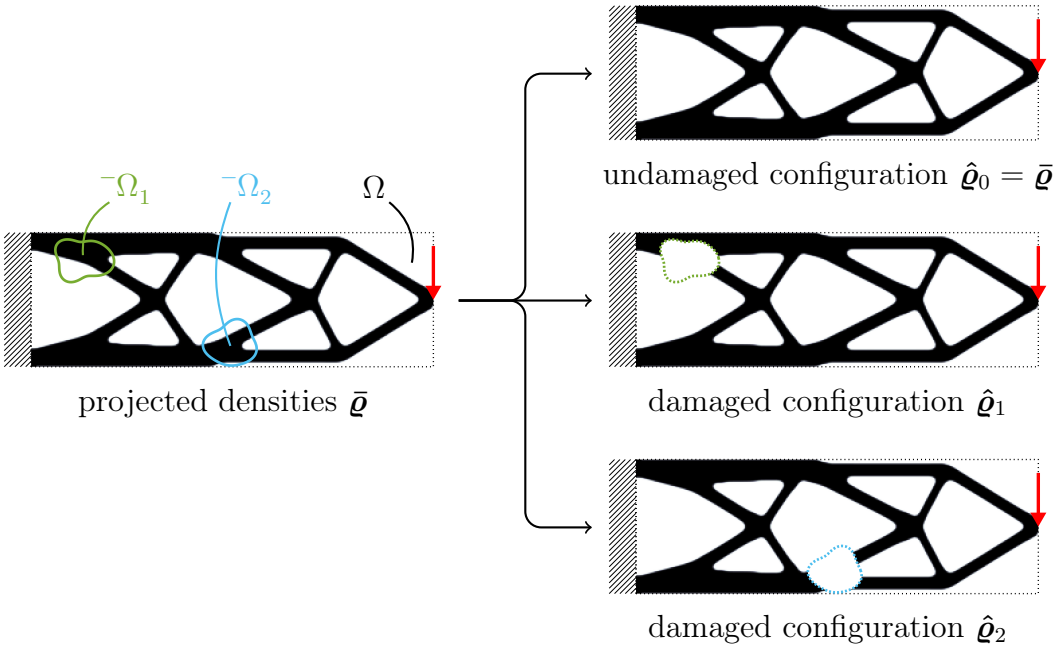


Figure 4.1 – Two exemplary damage zones ($^{-}\Omega_1$, $^{-}\Omega_2$) removed from the projected densities $\bar{\boldsymbol{\varrho}}$ of a cantilever beam structure, yielding the undamaged configuration $\hat{\boldsymbol{\varrho}}_0$ and two damaged configurations $\hat{\boldsymbol{\varrho}}_1$ and $\hat{\boldsymbol{\varrho}}_2$.

zone $^{-}\Omega_d$ (see figure 4.1). For the sake of completeness, also the undamaged configuration is considered as $\hat{\boldsymbol{\varrho}}_0 = \bar{\boldsymbol{\varrho}}$ or "scenario" $d = 0$:

$$\hat{\boldsymbol{\varrho}}_d = \hat{\boldsymbol{\varrho}}_d(\bar{\boldsymbol{\varrho}}, {}^{-}\Omega_d), \quad d = \{1, \dots, n_d\} \tag{4.1a}$$

$$\hat{\boldsymbol{\varrho}}_0 = \bar{\boldsymbol{\varrho}} \tag{4.1b}$$

The modified densities expand the workflow of the three field TO presented in section 2.3 by a fourth field. The modified densities now take on the role as the physical densities being used for the material interpolation scheme of the FE analysis discussed in section 2.3.4.

The concept of modified densities is further elaborated in section 4.3.1. In the later section 6.3.6, modified densities are even employed for a use beyond plain damage modeling in the context of assembly optimization.

4.1.2 Objective functions for damaged configurations

To quantify the effect of the individual damage scenarios, the studied objective function Φ needs to be evaluated for all considered configurations $\hat{\boldsymbol{\varrho}}_d$, each requiring an own FE analysis. The objective function for a damage scenario d is further abbreviated as Φ_d:

$$\Phi_d := \Phi(\hat{\boldsymbol{\varrho}}_d) = \Phi\left(\hat{\boldsymbol{\varrho}}_d(\bar{\boldsymbol{\varrho}}, {}^{-}\Omega_d)\right), \quad d = \{0, 1, \dots, n_d\} \tag{4.2}$$

4.1.3 Linking gradients of the damaged to the undamaged configuration

Gradients obtained during the FE analyses are now sensitivities with respect to the damaged configurations $\hat{\boldsymbol{\varrho}}_d$. However, by incorporating the partial derivatives $\frac{\partial \hat{\boldsymbol{\varrho}}_d}{\partial \bar{\boldsymbol{\varrho}}}$, the sensitivities can be

expressed with relation to the underlying undamaged field of projected densities $\bar{\boldsymbol{\varrho}}$:

$$\frac{\partial \Phi_d}{\partial \bar{\varrho}_i} = \frac{\partial \Phi_d}{\partial \hat{\varrho}_{d,i}} \frac{\partial \hat{\varrho}_{d,i}}{\partial \bar{\varrho}_i} \quad \forall d \tag{4.3}$$

This allows to further treat the gradients with the same workflow as in the standard three field TO approach. By incorporating the derivatives of the filter and projection equations, the sensitivities with respect to the original design variables $\boldsymbol{\varrho}$ are obtained (see Appendix D.7):

$$\frac{\partial \Phi_d}{\partial \varrho_i} = \sum_j \frac{\partial \Phi_d}{\partial \hat{\varrho}_{d,j}} \frac{\partial \hat{\varrho}_{d,j}}{\partial \bar{\varrho}_j} \frac{\partial \bar{\varrho}_j}{\partial \tilde{\varrho}_j} \frac{\partial \tilde{\varrho}_j}{\partial \varrho_i} \quad \forall d \tag{4.4}$$

The partial derivatives $\frac{\partial \hat{\boldsymbol{\varrho}}_d}{\partial \bar{\boldsymbol{\varrho}}}$ depend on the chosen damage model. For the two models presented in sections 4.2 and 4.3, the partial derivatives are summarized in Appendix D.8.

4.1.4 Optimization task for worst-case fail-safe optimization

Minimizing the objective function Φ_d with the maximum (or worst-case) value leads to a min-max problem taking into account all damage scenarios d. The objective then states (cf. [35]):

$$\min_{\boldsymbol{\varrho}} \max_{d} \Phi_d(\boldsymbol{\varrho}), \quad d = \{0, 1, \ldots, n_d\} \tag{4.5}$$

To be able to use gradient-based optimization algorithms, the max-operator needs to be approximated by a smooth, differentiable formulation. Suitable aggregation functions can be found in section 3. As explained in section 3.5.1, the KS aggregation function (A_{KS} from section 3.2.3) is used in this work for worst-case optimization tasks, yielding a conservative approximation of the maximum operator. The choice of the corresponding aggregation parameter p, which controls the properties of the aggregation function, is also discussed in section 3.5.1.

Using the KS aggregation function, the objective function for fail-safe optimization from equation (4.5) is approximated by:

$$\min_{\boldsymbol{\varrho}} \frac{1}{p} \log \left(\sum_{d=0}^{n_d} \mathrm{e}^{p \Phi_d(\boldsymbol{\varrho})} \right) \tag{4.6}$$

The complete optimization task is obtained by using the equations from (2.1) and exchanging the objective from (2.1a) by (4.6). The simplified data processing diagram for the optimization considering damage is shown in figure 4.2.

For the special case of a worst-case compliance minimization, the undamaged configuration for $d = 0$ does not have to be considered in the optimization. This saves one FE simulation per iteration. The compliance will increase for any removal of material since the gradient of the compliance has only negative entries. Therefore, the undamaged case will always have the lowest compliance of all considered failure scenarios and thus can never be the worst case. The optimization problem, considering the compliances c_d for the damaged configurations, then states:

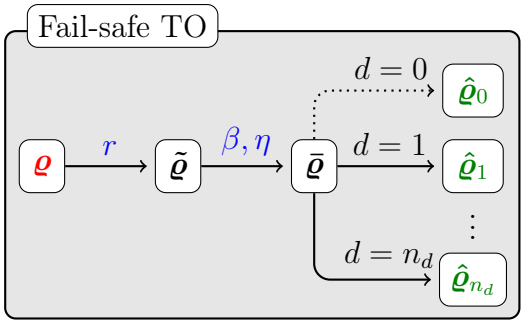


Figure 4.2 – Relation of variables for a TO considering damage.

$$\min_{\boldsymbol{\varrho}} \frac{1}{p} \log \left(\sum_{d=1}^{n_d} \mathrm{e}^{p c_d(\boldsymbol{\varrho})} \right) \tag{4.7a}$$

$$\text{s.t.} \quad h_i(\boldsymbol{\varrho}) \leq 0, \quad \forall i \tag{4.7b}$$

$$0 \leq \varrho_j \leq 1, \quad j = 1, \ldots, n_e \tag{4.7c}$$

$$\boldsymbol{K}(\boldsymbol{\varrho})\boldsymbol{u} = \boldsymbol{f} \tag{4.7d}$$

4.2 Damage model with fixed damage zones for regular meshes

A damage model involving rectangular damage zones is presented in section 4.2.1. This method is suited for regular FE meshes of equal element sizes.

Since the original method demands very high computational resources, two active-set strategies are discussed in sections 4.2.2 and 4.2.3 as a means to save computational cost by leaving out failure scenarios: A method based on geometrical considerations from [36] and the threshold method by the author, which was already published in [37]. While the first mentioned active-set strategy proved to lack robustness with respect to the disregarded failure scenarios, the latter strategy does not suffer from this problem.

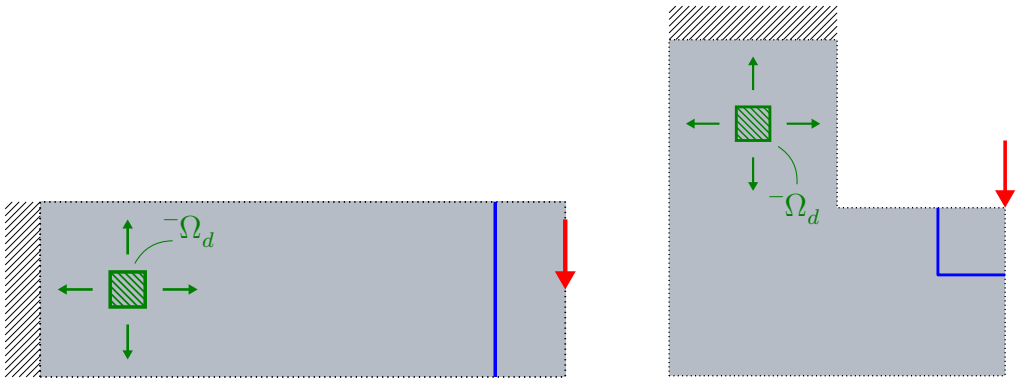
4.2.1 Simplified rectangular damage model

A method for fail-safe optimization of continua using a simplified damage model was first introduced by Jansen et al. in [35]. Their method uses rectangular shapes as damage regions $^{-}\Omega_d$ on a regular FE mesh of the design space.

At first, the admissible part of the design space inside which failure is considered is defined. Usually, this region covers the whole design space except for some buffer regions around the load introduction points of the structure and points where displacements are to be optimized.

Then, the size of the damage region is prescribed as an integral multiple of the element size in each dimension, resulting in a rectangular damage shape of constant size. The damage shape is placed at every possible location inside the admissible region until all locations are covered, as indicated in figure 4.3. This means, that the damage shape is moved across every dimension of the design space in discrete steps with a spacing of a single element size for neighboring damage regions. For each damage case, the set of affected element indices defining the damage region $^{-}\Omega_d$ are obtained on the regular mesh.

The modified densities $\hat{\boldsymbol{\varrho}}_d$ for every damage case d are obtained from the projected densities



(a) Cantilever beam example **(b)** L-bracket example

Figure 4.3 – Damage shape (green) used in the simplified rectangular damage model and border of the excluded region (blue) near the load for both studied examples.

$\bar{\boldsymbol{\varrho}}$ as follows:

$$\hat{\varrho}_{d,i} = \begin{cases} \bar{\varrho}_i, & \text{if element } i \notin {}^{-}\Omega_d \\ 0, & \text{if element } i \in {}^{-}\Omega_d \end{cases} \tag{4.8}$$

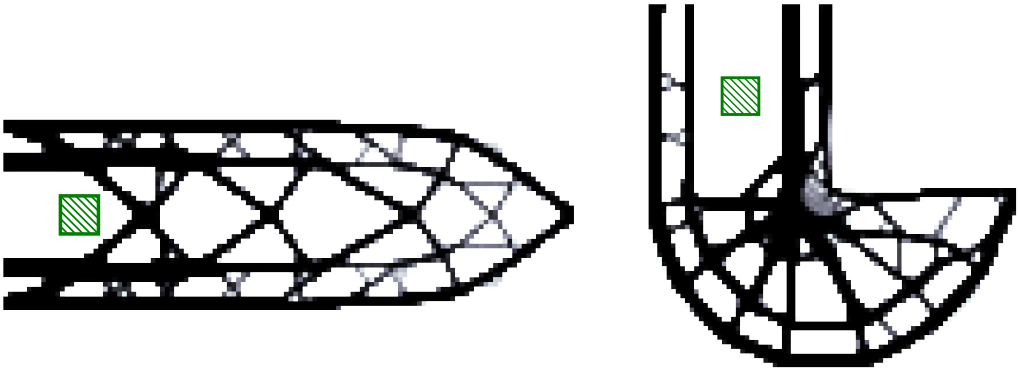
This corresponds to a one-to-one mapping if the elements are not deleted for the damage case considered and a total removal otherwise. The corresponding partial derivatives for obtaining the gradients as discussed in section 4.1.3 are given in Appendix D.8.

The drawback of this method of failure modeling is, that the total number of damage cases n_d is coupled to the discretization of the FE mesh. Therefore, the order of n_d is expected to be of the same order as the number of elements n_e. Since every damage case requires an own FE analysis, the computational effort is extremely high in this method, limiting its use to small 2D academic examples in relatively coarse discretizations.

Results for a fail-safe compliance minimization with a global volume constraint of $k_g = 0.4$ of the 2D cantilever and L-bracket problems, using the simplified rectangular damage model, are shown in figure 4.4. Due to the immense computational effort required for this method, the optimizations are only conducted for a drastically scaled-down version of the models, when compared to the exemplary designs shown in section 2.7. Otherwise, using the same discretization, involving $n_e = 30\,000$ elements, together with a damage shape size of 20×20 elements (one fifth of the height of the design space) and excluding a region twice as wide as a single damage shape from the loaded point (see figure 4.3) leads to $n_d = 19\,521$ damage scenarios for the cantilever and $n_d = 21\,161$ for the L-bracket example. Conducting again 200 iterations, the number of FE analyses required for the optimization totals to 3 904 200 or 4 232 200 analyses, respectively[11].

The results shown in figure 4.4 are thus obtained on a model with a discretization that is 2.5 times coarser per dimension than the corresponding designs shown in section 2.7. The number of elements is then $n_e = 4800$, the filter radius is $r = 1.6$, and the damage size is 8×8 elements. For the cantilever beam, $n_d = 3201$ damage cases exist, such that in total 640 200 simulations

[11] These high numbers of analyses correspond to a total duration of more than 45 days for every second that a single FE analysis takes, not taking into account the additional time for data processing and the optimization step itself. Therefore, applying this method demands massive use of parallelization and a lot of memory to store all gradients.



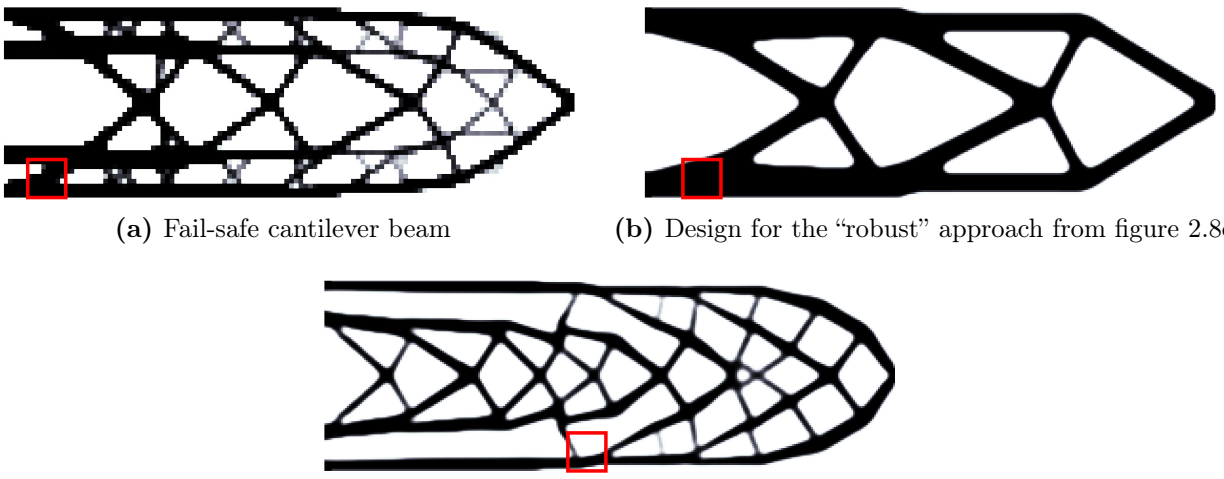
(a) Fail-safe cantilever beam **(b)** Fail-safe L-bracket

Figure 4.4 – Results obtained by the simplified rectangular damage model with indicated damage size (green shape) on a reduced resolution model.

are performed. For the L-bracket with $n_d = 3\,473$ damage cases, 694 600 analyses are conducted on the smaller model.

The designs from figure 4.4 contain several very thin features or areas with intermediate densities. Own numerical investigations show, that the convergence of a TO involving failure scenarios is much slower than for a plain TO, most probably due to "competing" gradients for the individual failure scenarios. This makes the optimization also sensitive to numerical round-off errors. Therefore, the cantilever beam result from figure 4.4a is slightly asymmetric, even though the geometry, loading, and distribution of damage shapes are not.

The locations of the damage shape for the worst-case failure scenarios for the cantilever beam are shown in figure 4.5. For comparison, also the corresponding worst-case damage for two results (with full resolution) from section 2.7.1, optimized without damage considerations, are shown. These include the design with the simplest topology obtained by the "robust" approach and the design with enforced redundancy, whose optimizations, with compliance as objective, both require only a computational cost comparable to a plain TO.



(a) Fail-safe cantilever beam **(b)** Design for the "robust" approach from figure 2.8c

(c) Redundant design from figure 2.8e

Figure 4.5 – Locations of the respective worst-case damages for the fail-safe and the "robust" and redundant cantilever beam designs from section 2.7.

The maximum (worst-case) compliance of all damage cases is referred to as $\bar{c}$ in the following sections:

$$\bar{c} = \max_{d} c_d, \qquad d = 1, \ldots, n_d \tag{4.9}$$

Table 4.1 summarizes the worst-case compliance values $\bar{c}$, the nominal compliances for the undamaged case c_0, and the volume fractions ω of the three studied designs. The values reveal, that by including damage considerations inside the optimization, the effect of a partial failure is drastically reduced for the fail-safe design. However, this comes at a price of very high computational cost.

Table 4.1 – Compliance values and volume fractions for the 2D cantilever beam example.

Design	Figure	$\bar{c}$	c_0	ω
Fail-safe	4.5a	372.34	249.01	40.00%
Robust approach	4.5b	12 157.13	213.68	40.02%
Redundant	4.5c	1 146.21	328.31	37.17%

The robustness of the fail-safe design from figure 4.5a with respect to the given failure is achieved by redundancy of the main load paths. This becomes easily visible near the BC region: A damage of the given shape is only able to fully cut away at most one of the four members connecting the part to the fixed boundary. Therefore, the fail-safe design suffers only from a relatively small increase in compliance, if this worst-case damage occurs, since then three load-bearing members stay intact.

The simple design from figure 4.5b does not offer these redundant load paths. A damage at one of the only two connecting members to the fixed boundary thus results in a massive increase of compliance for the damaged structure, as shown in table 4.1. However, the nominal compliance for the undamaged case is better than for the fail-safe design.

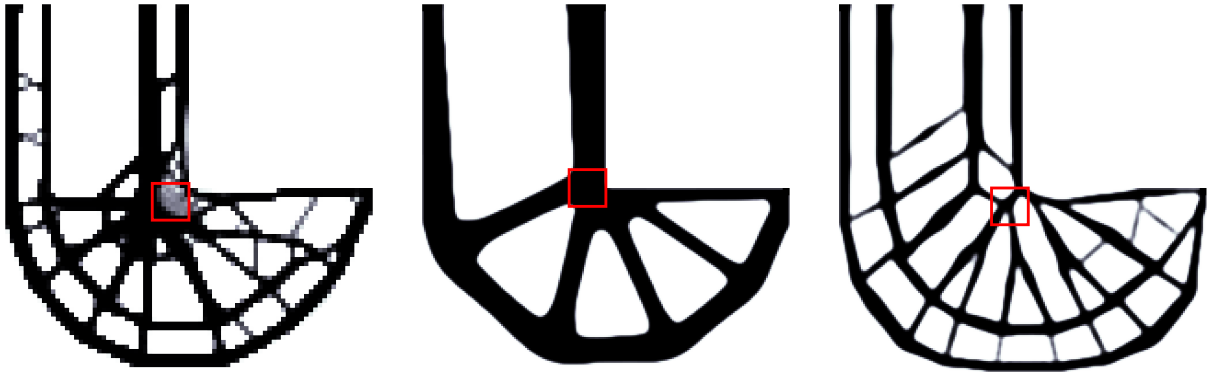
In contrast to that, the design with enforced redundancy from figure 4.5c offers implicitly increased robustness towards partial failure (see also [44]), since all thick main load paths are split into thinner, separated members. Although its worst-case compliance cannot compete with the fail-safe design, it is one order of magnitude better than the worst-case compliance of the simple design. However, the nominal compliance is the worst for the redundant design, but it also has the lowest volume.

The worst-case damage locations for the L-bracket example are shown in figure 4.6. Again, the design for the robust approach and the redundant design from section 2.7.2 are given for comparison. The compliance values are summarized in table 4.2.

Table 4.2 – Compliance values and volume fractions for the 2D L-bracket example.

Design	Figure	$\bar{c}$	c_0	ω
Fail-safe	4.6a	220.04	129.92	40.00%
Robust approach	4.6b	26 896.32	100.97	40.02%
Redundant	4.6c	4 695.18	152.37	36.26%

As before, the fail-safe design shown in figure 4.6a achieves robustness with respect to partial failure by incorporating redundant load paths. Moreover, the weak spot at the inner corner of the L-bracket is reinforced by a large accumulation of material.



(a) Fail-safe L-bracket

(b) Design for the "robust" approach from figure 2.9c

(c) Redundant design from figure 2.9e

Figure 4.6 – Locations of the respective worst-case damages for the fail-safe and the "robust" and redundant L-bracket designs from section 2.7.

The effect of damage is again highest for the non-redundant design from figure 4.6b, and, considering the worst-case compliance, the design with enforced redundancy shown in figure 4.6c is a midway between the fail-safe and the non-redundant design.

In contrast to the cantilever beam example, potentially critical areas for the L-bracket include not only the connecting members to the fixed boundary but also the region around the inner corner of the bracket. While the redundant design from figure 4.6c also offers four members connected to the BC region, just like the fail-safe design from figure 4.6a, it has only a poorly reinforced area in the vicinity of the corner. Thus, the effect of a damage at the corner is still relatively high for the design from figure 4.6c, as shown in table 4.2.

The results for the two studied academic 2D examples show, that, with the simplified rectangular damage model, the optimized topologies are clearly dependent on the chosen damage shape size. For sufficiently large damage sizes, robustness towards damage is achieved by the forming of redundant load paths. Too small damage sizes however may not lead to a redundant design, since the effect of the damage is then not the driving factor of the optimization and a result similar to a TO without damage consideration is obtained [35].

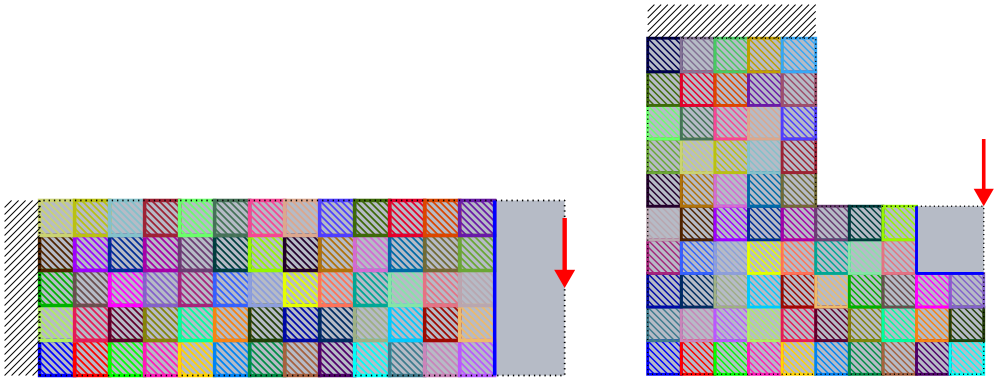
If redundant load paths are present in the fail-safe design, their separation distance typically resembles the chosen damage shape size. With this strategy, if one important load path is cut, the corresponding other load path will at least be partially intact and able to carry the load. In turn, this means, that even a slightly larger damage could be able to fully cut both load paths and lead to an unpredictable degradation of performance. The optimized designs are therefore in general sensitive to the size of the actual applied damage and an analysis of the effect of one damage size cannot be used to predict the effect of differently sized damage regions. However, for compliance as objective, a larger damage region will always have a worse effect of degradation than a smaller sized one, such that a result is obtained, that is at least robust with respect to smaller damage sizes compared to the size considered during the fail-safe optimization.

4.2.2 Preselection of damage scenarios by geometrical considerations

The high number of analyses needed for a fail-safe optimization with the simplified rectangular damage model from section 4.2.1 originates from the fact, that the corresponding damage shapes are placed with the smallest possible discrete separation distance of one element size in relation to each other. A straightforward idea to cut down the number of considered failure scenarios is

to increase the spacing of adjacent damage regions, as proposed in [36].

Following the approach from [36], the coarsest spacing of damage regions without leaving gaps is a spacing equal to the damage shape size. Then, also no overlap is present. In the following, this is referred to as grid-level 1 (GL1, or PA_1 in the original paper). The corresponding distribution of damage shapes is shown in figure 4.7.



(a) Cantilever beam example **(b)** L-bracket example

Figure 4.7 – Distribution of the damage shapes for GL1.

For all higher grid-levels, the spacing of the damage shapes is halved, such that also the middle positions between the shape positions of the next coarser grid-level are populated. Thus, e.g. for GL2 and GL3, the spacing is half or quarter the edge length of the damage shape, respectively.

While the GL1 spacing guarantees that every element in the admissible damage region is cut away exactly once when considering all damage cases, the obtained designs are very vulnerable to damage configurations with slightly offset damage positions. Obviously, areas of less penalized material exist, which then attract solid material during the optimization. The reason for this is, that members gathering around the edges of the considered damage zones are always just partially cut. This fact is analyzed in [36] from a geometrical point of view and a measure for the maximum "survival rate" depending on the grid-level is given. GL2 is then considered as sufficient to capture all critical damage scenarios if the damage shape is at least twice the size of the cut features.

While this approach clearly cuts down the computational cost for the corresponding optimization considering damage, in [36] there is no proof given, that the proposed reduction of damage scenarios is a conservative approach. Therefore, own numerical investigations in this matter were conducted in [37], an extended examination is also given in the following.

The cantilever beam and the L-bracket are studied with grid-levels of GL1, GL2, and GL3, respectively. In contrast to [36], where the obtained final results are only visually compared, here also the cross-validation with the corresponding other grid-levels, as well as for the full set of damage scenarios is performed.

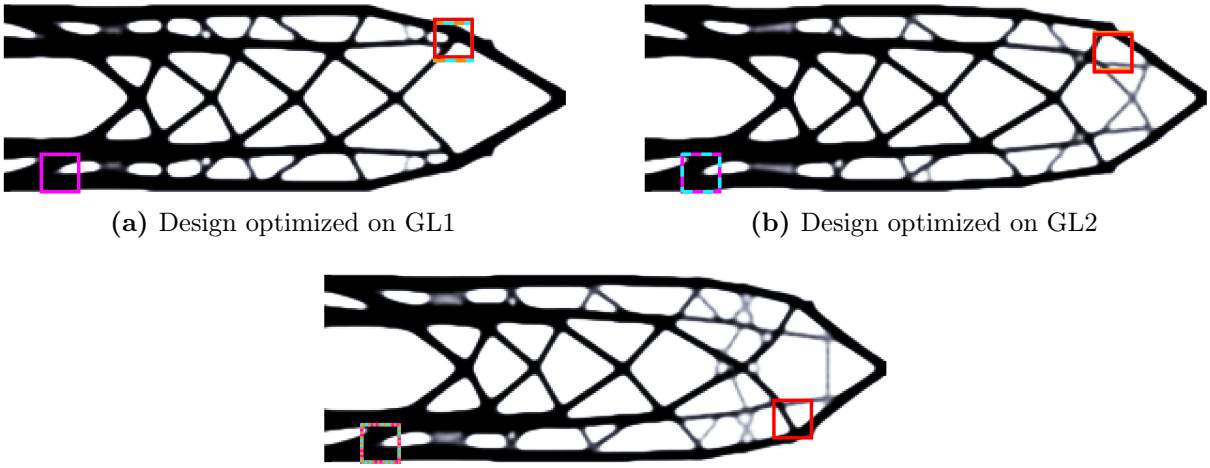
The reduction in computational cost allows to go back to the original discretization, as used in the examples in section 2.7. With a damage size of 20×20 elements (cf. figure 4.3), GL1, GL2, and GL3 correspond to a damage shape spacing of 20, 10, or 5 elements, respectively, while the full set has the minimum spacing of one element size. The number of damage scenarios in relation to the chosen grid-level is given in table 4.3.

The obtained designs for the cantilever beam, optimized on GL1, GL2, or GL3, respectively, are shown in figure 4.8. The locations of the worst-case damages of the four damage sets for

Table 4.3 – Number of damage scenarios for different damage sets for the cantilever beam and the L-bracket example.

Damage set	Cantilever	L-bracket
GL1	65	71
GL2	225	245
GL3	833	905
Full	19 521	21 161

cross-validation (GL1, GL2, GL3, and the full set) are indicated for every design. Table 4.4 lists the numerical values of the worst-case compliances ($\bar{c}^1$, $\bar{c}^2$, $\bar{c}^3$, and $\bar{c}^f$ for GL1, GL2, GL3, and the full set, respectively), nominal compliances c_0 and global volume fractions ω.



(a) Design optimized on GL1

(b) Design optimized on GL2

(c) Design optimized on GL3

Figure 4.8 – Results for the cantilever beam optimized with a reduced number of failure cases together with their respective worst-case damages on GL1 (magenta), GL2 (cyan), GL3 (orange), and the full set (red).

Table 4.4 – Compliance values and volume fractions for the 2D cantilever beam example with reduced damage cases.

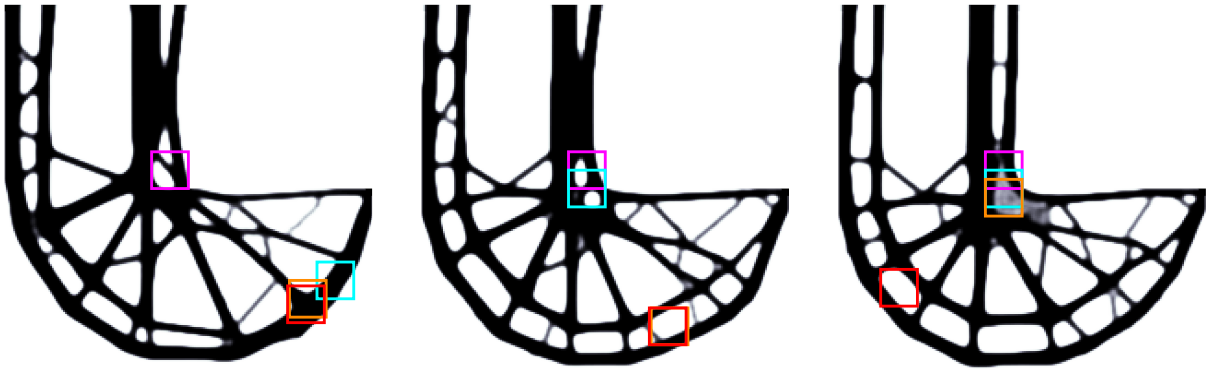
Design	Figure	$\bar{c}^1$	$\bar{c}^2$	$\bar{c}^3$	$\bar{c}^f$	c_0	ω
GL1	4.8a	359.34	4 590.74	4 590.74	4 590.75	249.57	40.00%
GL2	4.8b	370.82	370.82	1 268.12	1 268.12	258.98	40.00%
GL3	4.8c	378.16	378.16	378.16	1 196.38	258.70	40.00%

Although the designs from figure 4.8 look very similar to each other, a large interval of worst-case compliance values is reported in table 4.4. Low worst-case values are only obtained up to the grid-level that was considered during the optimization, also coarser grid-levels are included in the finer ones. However, in all shown cases, critical damage positions were skipped when validating the result on a finer grid-level or the full set. Regarding the values in table 4.4 and the worst-case damage positions in figure 4.8, neither the effect nor the location of the true

worst case considering the full set of damage cases could be captured or predicted by using reduced damage sets.

As already discussed before, GL1 is not sufficient for obtaining a fail-safe design, hence the effect of the skipped damage cases is the worst for this design. Interestingly, even for the GL3 design, still an unpredictable increase of worst-case compliance is observed considering the full set. In fact, the GL3 design has a worse performance under damage on the full set than the redundant design in figure 4.5c, which is not even optimized considering damaged states and has less volume (cf. tables 4.1 and 4.4).

For the L-bracket example, similar observations as for the cantilever beam are made. The optimized designs for the three considered grid-levels are shown in figure 4.9 together with the worst-case damage locations. The worst-case compliance values are summarized in table 4.5. Again, coarser damage sets cannot be used to estimate the location or effect of a damage on a finer damage set. Additionally, for the L-bracket example, it is observed that the areas of intermediate densities near the inner corner increase in size, the finer the grid-level becomes and approach the area of the result from figure 4.6a.



(a) Design optimized on GL1 **(b)** Design optimized on GL2 **(c)** Design optimized on GL3

Figure 4.9 – Results for the L-bracket optimized with a reduced number of failure cases together with their respective worst-case damages on GL1 (magenta), GL2 (cyan), GL3 (orange), and the full set (red).

Table 4.5 – Compliance values and volume fractions for the 2D L-bracket example with reduced damage cases.

Design	Figure	$\bar{c}^1$	$\bar{c}^2$	$\bar{c}^3$	$\bar{c}^f$	c_0	ω
GL1	4.9a	184.85	1 990.66	2 008.32	2 008.35	129.22	40.00%
GL2	4.9b	194.95	205.38	822.81	875.65	135.45	40.00%
GL3	4.9c	203.16	207.95	216.56	284.78	137.99	40.00%

The cross-validations conducted in [37] and this work reveal, that TO problems involving the simplified rectangular damage model from [35] together with the damage case reduction method proposed by [36] are very sensitive to the exact placement of the considered damage zones. Therefore, the purely geometrically derived method from [36] is not a conservative approach for the reduction of the computational cost of a fail-safe optimization. For the same reason, also interpolation schemes or surrogate models can be ruled out as means to reduce the number of analyses conducted during each optimization iteration. Additionally, the discussed damage case

reduction methods inherit the sensitivity towards the size of the damage shape from the original method.

4.2.3 Preselection of damage scenarios by the threshold method

Another approach for reducing the overall number of damage scenarios to consider during optimization of a fail-safe part is to adaptively preselect potentially critical damage cases within each iteration and disregard non-critical damage cases using an active-set strategy. Nevertheless, to quantify the effect of a failure scenario reliably, a dedicated FE simulation is needed. Perturbation techniques, e.g. relying on the sensitivities inside a damage zone, proved to be incapable to predict the importance of the material to be removed (see e.g. [35, 102]). Therefore, in order to save computational effort, heuristic approaches for preselection of damage cases have to be used.

A straightforward measure for the criticality of a damage case is already given in the original paper introducing the simplified rectangular damage model in [35]: Empty damage zones that do not contain any solid material obviously do not need to be considered as a damage case. However, within a TO, truly empty regions may only exist in a well-converged state, meaning, this preselection mechanism only has an effect at a stage, where the optimization is about to finish.

To overcome this problem, the threshold method by the author [37] relaxes the condition for a damage zone to be considered as "empty". A threshold value $\bar{\varrho}_t$ is applied to the maximum projected variables inside each damage zone. If these variables all fall below the threshold value, the corresponding damage scenario is not considered during the current optimization iteration.

The idea behind this heuristic approach is, that if a certain region of the design space is critical for the load-bearing capability of the structure, it will quickly converge to fully solid material and then automatically be considered as a potentially critical damage zone. Conversely, if a region contains only elements of low or intermediate density, it is most probably not vital for the structural performance of the part.

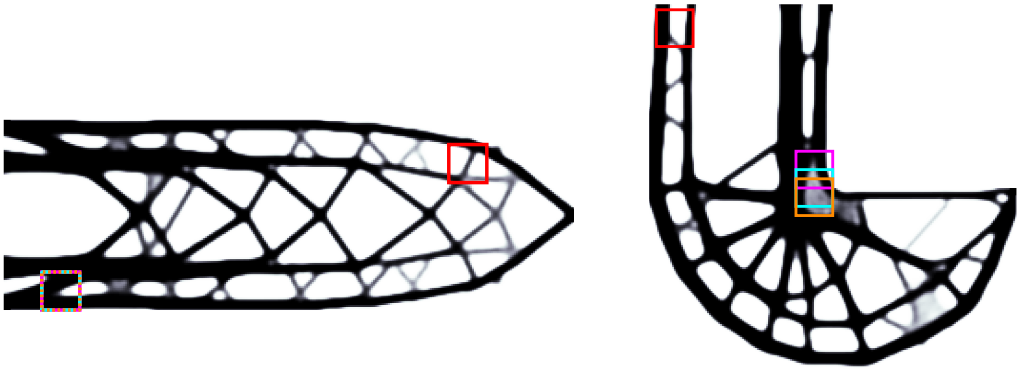
Numerical results presented in [37] show, that the threshold value can be set to values as high as $\bar{\varrho}_t = 0.9$ with only little effect on the worst-case compliance of the optimized part.

Results for the cantilever beam and the L-bracket optimized using the threshold method with $\bar{\varrho}_t = 0.9$ are shown in figure 4.10, where also the worst-case damage locations of the cross-validation are marked. Due to computational limitations, the optimizations were conducted on grid-level GL3 (cf. section 4.2.2).

The corresponding compliance values are listed in table 4.6. The values underline, that by using the threshold method, no critical damage scenarios of the considered damage set are skipped. Up to the used grid-level GL3, similar worst-case performances are achieved with the threshold method, as without thresholding. In fact, for the cantilever beam result from figure 4.10a, the worst-case compliances have even improved, despite a notably stronger asymmetry of the design, when compared to the design from figure 4.8c.

Table 4.6 – Compliance values and volume fractions for the results obtained by the threshold method with $\bar{\varrho}_t = 0.9$.

Design	Figure	$\bar{c}^1$	$\bar{c}^2$	$\bar{c}^3$	$\bar{c}^f$	c_0	ω
Cantilever beam	4.10a	377.62	377.62	377.62	767.54	264.43	40.00%
L-bracket	4.10b	210.07	212.84	221.43	452.99	138.80	40.00%



(a) Cantilever beam **(b)** L-bracket

Figure 4.10 – Results obtained by the threshold method with $\bar{\varrho}_t = 0.9$ on GL3, together with their respective worst-case damages on GL1 (magenta), GL2 (cyan), GL3 (orange), and the full set (red).

When using the threshold method, the computational saving is highest in the beginning of the optimization and decreases towards the end, since then many structural features distributed across the whole design space have evolved, such that "empty" damage regions become rare. Figure 4.11 shows the percentage of the actually considered damage cases from the damage set GL3 for each optimization iteration of the shown examples. The total number of FE analyses conducted during the whole optimization is summarized in table 4.7.

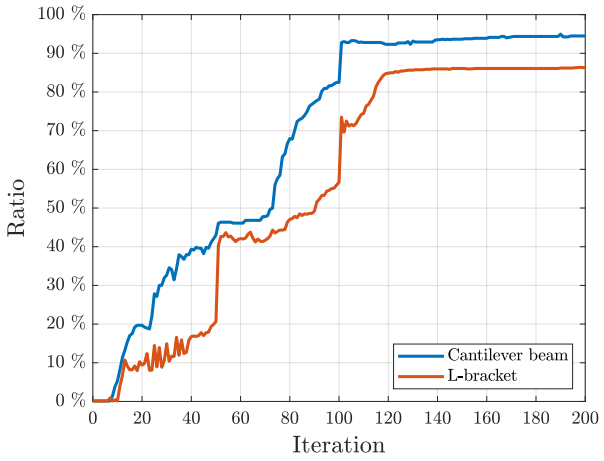


Figure 4.11 – Percentage of the considered damage scenarios when using the threshold method.

Table 4.7 – Total number of FE analyses for an optimization with 200 iterations considering damage on GL3 with or without the threshold method.

Example	w/o threshold	with $\bar{\varrho}_t = 0.9$
Cantilever beam	166 600	113 392 (68%)
L-bracket	181 000	101 680 (56%)

By using the threshold method, the computational cost is cut down to 68% or 56%, respectively, which means, that it remains at a very high level. Therefore, the optimizations

are conducted only on the reduced set GL3 of damage scenarios and not on the full set. As a consequence, the true worst-case damage scenario involving the full set could again not be retrieved (see figure 4.10), which is a persisting problem originating from the use of a sparser grid-level for the damage distributions, as explained in section 4.2.2. If the full set of damage scenarios would be used within the optimization and a similar relative saving is achieved by the threshold method, the optimization would still require an infeasible amount of time. Therefore, in order to be suitable for industrial application, a more severe reduction of several orders of magnitude concerning the number of FE analyses is needed for fail-safe optimizations.

The presented threshold method reduces the calculation cost of a fail-safe optimization in disregarding empty damage zones. Still, a large drawback of the used simplified rectangular damage method remains: The number of damage cases is still coupled to the discretization of the underlying FE mesh. Hence, a large number of damage cases are analyzed that eventually cut the same structural feature of the design, but only at different positions, resulting in dispensable analyses. Considering e.g. a beam-like feature, a single cut through the full cross-section at any position would be sufficient to capture the effect of the failure of said feature. Following this idea leads to a load-path-based damage model with adaptively shaped damage zones, discussed in the following section 4.3.

4.3 Damage model with parametric damage zones for arbitrary meshes

To further cut down the number of damage scenarios to consider in a fail-safe optimization, a load-path-based method from the author, presented in [58], is used. This method involves damage zones of adaptively chosen size and shape.

The basic idea is to move away from tiling the whole design space with damage zones of constant shape and size, as in the methods discussed in section 4.2, and rather cover all load-bearing members of the part by specifically tailored damage shapes. This allows covering possible critical damage scenarios ideally with a minimal amount of damage shapes, namely one damage shape per load-bearing feature. Then, all load paths are cut once (and only once), which brings the fail-safe optimization problem closer to the intention of the CS-25 [14], which demands a fail-safe structure to resist the designated loads, even if one load path fails. At the same time, the computational effort for the involved optimization is drastically reduced.

This approach is thus slightly different than the methods discussed in section 4.2, in which, depending on the chosen size of the damage zones, situations may exist where either no load-bearing member is cut completely or, in the other extreme, some damage zones may cut away several load-bearing members at once.

The presented framework for load-path-based damage scenarios also remedies the main drawbacks of the rectangular damage model from [35]: The total number of damage scenarios becomes independent of the FE discretization and is only dictated by the number of load-bearing features being present. Also, robustness with respect to the exact positioning and size of the damage zones is achieved. Lastly, the load-path-based method is also applicable to models with unstructured FE meshes.

To achieve the mentioned improvements, the first step is to adapt the workflow involving modified densities $\hat{\boldsymbol{\varrho}}$, presented in section 4.1.1, to a use with arbitrarily shaped damage zones. This is discussed in section 4.3.1. Section 4.3.2 lays the mathematical foundation for a use of parameterized damage zones, described by analytical functions. This allows to place differently sized and shaped damage shapes across the design space, which is demonstrated in section 4.3.3. The extraction of the required parameters by automated methods is discussed in section 4.3.4.

Finally, section 4.3.5 discusses the application of load-path-based damage directly in a TO and motivates the use of an alternative type of optimization, which is discussed in section 5.

4.3.1 Generalization of damage zones by damage masks

A systematic way to describe a damage region is introduced. In order to describe which elements are deleted and which stay unchanged, a multiplicative mask $\bar{}\boldsymbol{\psi}_d$ for every damage scenario d is incorporated. The final physical density layouts are then the modified densities $\hat{\boldsymbol{\varrho}}_d$ obtained by taking the Hadamard (entry-wise) product of the projected densities $\bar{\boldsymbol{\varrho}}$ and the damage mask vector $\bar{}\boldsymbol{\psi}_d$ for every damage case d [58]:

$$\hat{\boldsymbol{\varrho}}_d = \bar{\boldsymbol{\varrho}} \circ \bar{}\boldsymbol{\psi}_d, \quad \forall d \tag{4.10a}$$

$$\bar{}\psi_{d,i} \in [0,1] \tag{4.10b}$$

The entries of $\bar{}\boldsymbol{\psi}_d$ are values in the interval from 0 to 1, with 1 meaning that the structure is left unchanged and 0 meaning that material is completely removed in this area[12]. With this definition, also partial removal is possible when using values between 0 and 1. An exemplary application of a damage mask is shown in figure 4.13.

The main advantage of this definition is, that now parametric geometrical shapes can be projected onto a mask vector and then be used as damage shapes even on irregular meshes, as discussed in the following sections 4.3.2 and 4.3.3.

4.3.2 Projecting parametric damage shapes onto damage masks

In this work, superellipses (in 2D) or superellipsoids (in 3D) are used to model the shape of possible damages to a structure, as presented by the author in [58, 101]. These shapes are very flexible and allow e.g. a transition from a round to an almost box-shaped geometry by altering a shape parameter. The following equations are given for a 3D superellipsoid. For the 2D case, the z-coordinates and corresponding size parameters in the z-direction can simply be omitted.

The goal is to project the shape's geometry onto the FE mesh and thus to obtain a damage mask vector $\bar{}\boldsymbol{\psi}$ (the index d is omitted for now) that will create a cut of the specified shape to an existing structure by removing material that resides inside the outline of the shape.

Firstly, to analytically describe the location, orientation, size, and shape of the superellipsoid, the following parameters need to be given (see figure 4.12):

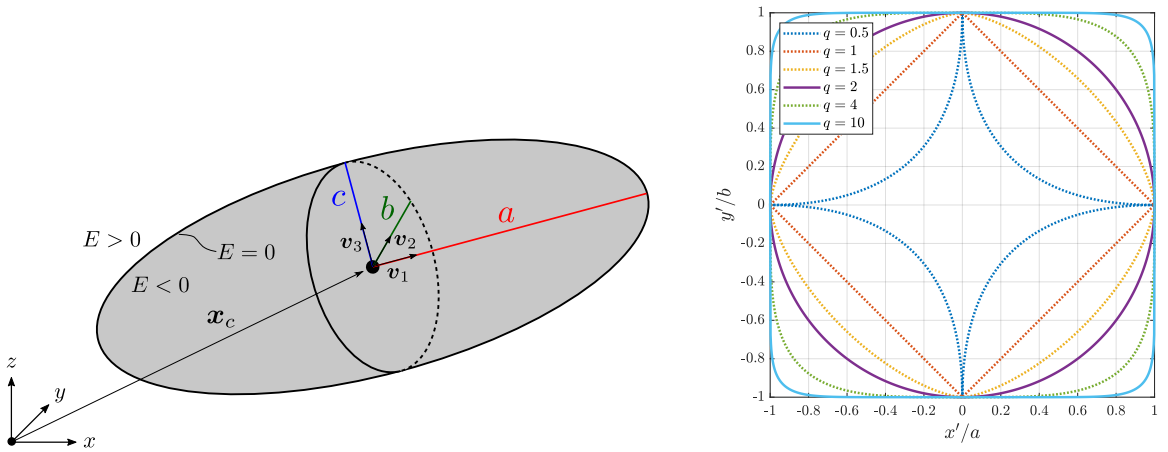
- The center point's location vector $\boldsymbol{x}_c$ in global coordinates,
- an orthogonal matrix $\boldsymbol{V}$, defining the principal axes directions in global coordinates,
- a size vector $\boldsymbol{a} = [a, b, c]^T$, defining the lengths of the semi-axes,
- the shape exponent q.

The matrix $\boldsymbol{V}$, together with the center point's location vector $\boldsymbol{x}_c$, is used to transform position vectors $\boldsymbol{x}$ from the global coordinate system to a coordinate vector $\boldsymbol{x}'$ in the shape's local coordinate system:

$$\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3] \tag{4.11}$$

$$\boldsymbol{x}' = \boldsymbol{V}^T(\boldsymbol{x} - \boldsymbol{x}_c) \tag{4.12}$$

[12] Formally, the definition from equation (4.10a) also covers the simplified damage model from section 4.2.1 with the modified densities given in equation (4.8). The mask $\bar{}\boldsymbol{\psi}_d$ is then simply a binary vector with ones in regions outside the damage rectangle and zeros inside.



(a) Parameters of an ellipsoid ($q = 2$)

(b) Normalized shape outlines ($E = 0$) for different q in the x'-y'-plane

Figure 4.12 – Parametric geometry of a superellipsoid.

In local coordinates $\boldsymbol{x}' = [x', y', z']^T$, the outline of the superellipsoid is defined as the isosurface $E = 0$ of following function (cf. figure 4.12):

$$E(\boldsymbol{x}') := \left|\frac{x'}{a}\right|^q + \left|\frac{y'}{b}\right|^q + \left|\frac{z'}{c}\right|^q - 1 = 0 \tag{4.13}$$

Here, q is the exponent to alter the shape of the outline, as demonstrated in figure 4.12b. In this work, only convex shapes with values of $q = 2$ and $q = 10$ are used.

Finally, the above analytical expressions are used to obtain a mask value for every finite element of the discretized design space of the structure. With the definition from equation (4.13), a positive scalar $E > 0$ is obtained for all coordinates that lie outside of the shape. These are the regions that should not be affected by the damage shape and thus should be projected towards a mask value of 1. Conversely, regions inside the damage shape with $E < 0$ should be mapped to 0. As a differentiable approximation to the Heaviside step-function, a sigmoid function involving the hyperbolic tangent is used to model smooth mapping characteristics. A parameter α controls the sharpness of the transition between 0 and 1 and is set to $\alpha = 10$ in this work. The damage mask value for element i, with its center point located at $\boldsymbol{c}^i$, is calculated as:

$$^{-}\psi_i = \frac{\tanh(\alpha E(\boldsymbol{c}^{i'})) + 1}{2} \tag{4.14a}$$

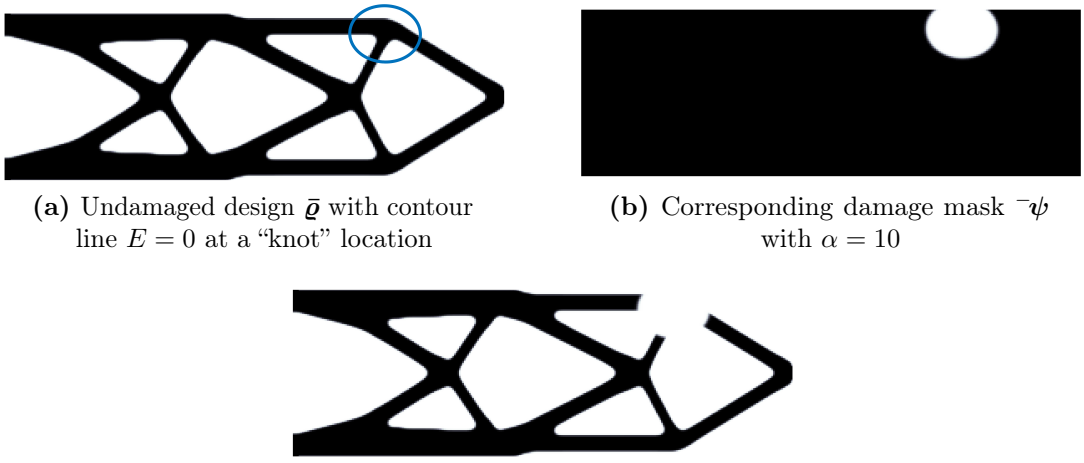
$$\boldsymbol{c}^{i'} = \boldsymbol{V}^T(\boldsymbol{c}^i - \boldsymbol{x}_c) \tag{4.14b}$$

Since the entries of $\boldsymbol{x}_c$ in the above equation are own, continuously variable parameters that are independent of the elements' center point coordinates $\boldsymbol{c}^i$, the location of the damage shape can now also vary continuously and independently of the discretization of the underlying FE mesh.

4.3.3 Applying parametric damage to load-bearing parts

With the mathematical foundation of the previous sections 4.3.1 and 4.3.2, adaptively sized and oriented damage shapes can be defined, that cover all load-bearing parts of a structure [58, 101].

An example of applying a single elliptical damage patch to a 2D structure by the described method is shown in figure 4.13. While it is possible to manually place damage shapes on the structure, more convenient automatic methods for parameter extraction exist, as discussed in the following section 4.3.4.



(a) Undamaged design $\bar{\boldsymbol{\varrho}}$ with contour line $E = 0$ at a "knot" location

(b) Corresponding damage mask $\bar{}\boldsymbol{\psi}$ with $\alpha = 10$

(c) Resulting damaged design $\bar{\boldsymbol{\varrho}} \circ \bar{}\boldsymbol{\psi}$

Figure 4.13 – Example of applying damage to a structure by a multiplicative mask $\boldsymbol{\psi}$ generated from a parameterization E of an ellipse.

4.3.4 Automated load path identification

Due to the automated methods available, this work focuses on the load path identification of lattice-type structures. For 2D problems with low volume fractions, these are mostly automatically obtained. For 3D problems, often plate-like structures are optimal. However, when enforcing redundant designs by utilizing local volume constraints (see section 2.6.2), even for 3D problems lattice-type structures can be obtained.

The definition of damage scenarios on load-bearing features of a structure consists of these steps:

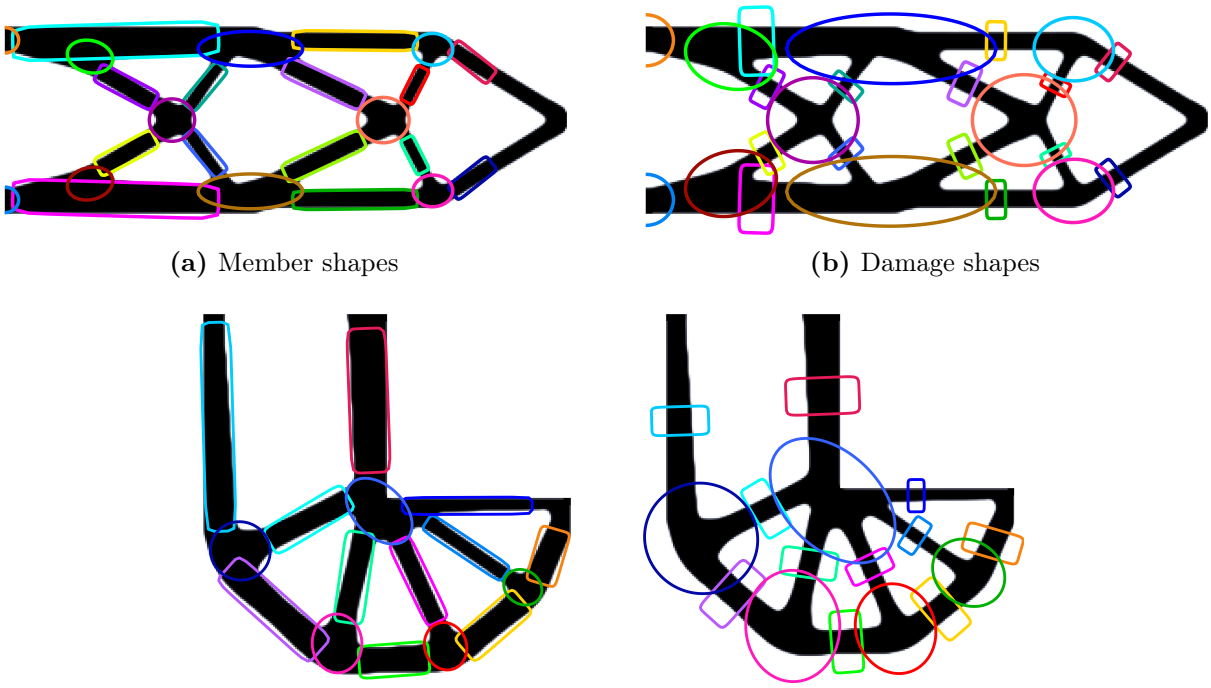
1. Identification of load-bearing members
2. Analytical description of individual members by auxiliary "member shapes"
3. Conversion of "member shapes" to "damage shapes"
4. Generation of damage mask vectors $\bar{}\boldsymbol{\psi}_d$

In the first step, load-bearing members are identified by subdividing the solid phase of a topology optimized design into "beams" and "knots". Beams are slender, highly directional parts, whereas knots are the regions where several beams meet. Particularly for lattice-type structures, automated processes exist for this subdivision. Appendix B.1 describes the details of two methods successfully applied by the author based on a simple stress criterion and image processing algorithms.

After the subdivision is done and corresponding parameters are calculated (see Appendix B.1), in a second step, auxiliary "member shapes" are constructed for each identified member: Beams are expected to be slender members of constant thickness which could best be described by rectangular boxes, whereas knots are aggregations of material of varying shapes which

are approximated by ellipsoids. The parametric superellipsoids presented in section 4.3.2 can describe both types of shapes.

As an example, automatically extracted member shapes using the stress criterion method for the "robust" cantilever beam and L-bracket designs are shown in figure 4.14a and 4.14c. With a superellipse shape parameter $q = 2$ for knots and $q = 10$ for beams (cf. equation (4.13)), the member shapes describe the solid load-bearing members of the parts adequately well. However, an area around the load is again excluded from the consideration, since later no damage will be applied directly at the loads (cf. figure 4.3).



(a) Member shapes **(b)** Damage shapes

(c) Member shapes **(d)** Damage shapes

Figure 4.14 – Member shapes and corresponding damage shapes for the "robust" cantilever beam and L-bracket examples as an overlay on the projected density fields.

As a third step, the member shapes are transformed to damage shapes. This involves an up-scaling of the member shapes, such that the damage shape is guaranteed to fully enclose the corresponding original member with some excess size. Optionally, the damage shapes for beams can be rendered more compact to represent only a cut through the middle of the corresponding beam. For all shown examples, a cut through a beam's whole cross-section at any location is sufficient to fully degrade the load-bearing capability of the beam (for a discussion when this simplification is applicable, see Appendix B.2). Therefore, these compact damage shapes for beams are used throughout this work.

Following this approach, figures 4.14b and 4.14d show the obtained damage shapes, involving an up-scaling of all corresponding member shapes by a factor of 2 and a subsequent compactification of the shapes for beams by modifying the semi-major axis (length direction) to be half the size of the semi-minor axis (width direction), i.e. $a = 0.5b$ (cf. figure 4.12a).

Finally, as the fourth step, with the size and orientation parameters of the damage shapes at hand, a damage mask vector $\bar{}\boldsymbol{\psi}_d$ is calculated for every shape d, according to equations (4.13) and (4.14). Then, a fail-safe optimization with load-path-based damage can be conducted.

4.3.5 Using load-path-based damage directly in fail-safe TO

A result for a fail-safe TO involving a load-path-based damage model was presented by the author in [37]. Therein, a compliance minimization of a cantilever beam under partial damage, subjected to a global volume constraint, was studied. The obtained result is repeated in figure 4.15. In this earlier version of the load-path-based damage model, the load-bearing members were cut directly according to their current shape and no approximation by auxiliary damage shapes was done.

Figure 4.15 – Result for a fail safe cantilever beam from [37], obtained by TO using load-path-based damage directly.

The automated load path identification discussed in section 4.3.4 is performed during the TO, meaning that also intermediate designs are analyzed and decomposed into beams and knots. As reported in [37], the computational effort of the load-path-based fail-safe TO is comparable to an optimization involving the rectangular damage model with a reduction to GL1 (cf. section 4.2.2). The reason for this is, that, with the load-path-based damage model, all solid members are covered once and no overlap of damage zones is needed. However, in contrast to an optimization involving damage on GL1, the risk that material gathers in less penalized areas and thus critical damage scenarios are skipped is eliminated, since the damage shapes are always adapted to the current location of the load-bearing features.

Unfortunately, the adaptation of the damage zones to the current design is also the source for the major drawback of a load-path-based fail-safe TO, as outlined in [37]: The optimization problem is changed unsteadily each time a new set of damage zones is identified. This negatively affects the convergence behavior of the problem: If e.g. a load path evolves and is identified as such, it will in the following iterations be penalized by damage. Therefore, this load path may become less attractive for the optimizer and is eventually replaced by another load path, until this one is also recognized and then cut away. This effect may either settle or in some cases be so strong that convergence to a final design cannot be guaranteed and oscillations in the design occur. Another drawback is, that there is no tuning parameter offering the user control over desired properties of the optimized design, like e.g. the degree of redundancy or the spacing of redundant load paths.

The following section 5 remedies the above-mentioned problems by introducing a two-stage optimization procedure. This procedure allows to tackle, among others, fail-safe optimization problems involving load-path-based damage in a numerically stable way.

5 Density-based shape optimization

The presented method for a shape optimization (SO) is based on a two-step procedure involving a TO as the first step to generate a preliminary design with a well-defined geometry and a subsequent SO as the second step, which then optimizes the outlines of the design. Such sequential optimization techniques have been applied successfully before (see e.g. [103–105]).

In a SO, the contour of a part is altered to minimize an objective function. Typically, the shape of the part is described by a set of parameters (e.g. control points) that are used as the design variables of the problem. The part is meshed with a conforming mesh with nodes on the part's boundary. During the optimization, the outer nodes are moved according to a mapping function between the design variables and the nodal positions (see e.g. [106–108]).

In contrast to that, the density-based SO presented in this work uses the same FE mesh as for the TO, therefore no remeshing nor data conversion or mapping is needed if a design originating from a TO is processed. Additionally, the proposed SO method adapts the variable processing steps of the three field TO approach discussed in section 2, which eases its implementation in existing TO codes. Most parts of this section are taken from a preceding work of the author, which is published in [58].

The basic theory of the presented two-stage optimization method, involving a density-based SO, is discussed in section 5.1. The main application of the SO method in this work is the optimization of fail-safe structures. As shown in section 4.2.1, topology optimized designs with enforced redundancy have the potential to be relatively robust towards partial damage. However, since failure is not explicitly considered during their optimization, there is still room for improvement. The basic idea followed in this work is to treat these designs as preliminary designs and further optimize their shape towards a fail-safe design, which is studied in section 5.2.

Numerical examples shown in section 5.3 confirm a rapid convergence of the SO stage for fail-safe applications. Section 5.4 discusses the computational cost and possible further efficiency improvements for a SO-based fail-safe optimization when compared to the alternative methods from section 4, involving solely a TO stage.

Section 5.5 presents examples for an alternative optimization formulation, sticking closer to the definition of fail-safety of structures. Finally, in the last section 5.6, other possible use cases for density-based SO are briefly discussed.

5.1 Theory of density-based shape optimization

A SO can only be performed, if some initial geometry is already present. Therefore, it is suggested to first conduct a TO using the three field TO discussed in section 2 as first step or "stage I". Afterward, the proposed density-based SO is performed as "stage II". In the following, the superscripts $(\cdot)^I$ or $(\cdot)^{II}$ are included to clarify if variables originate from the preliminary design of stage I or from the SO of stage II discussed in this section.

Figure 5.1 shows the relation of variables for the simplest case of a plain TO in stage I and a plain SO in stage II and is intended as a help to keep track of which variables are used in the individual optimization stages. To underline the strongly sequential relation, the TO stage in figure 5.1 is offset to the background, indicating that the TO of stage I has to be finished in order to start the SO of stage II. Only the filtered variables $\tilde{\boldsymbol{\varrho}}^I$ from the final iteration of stage I serve as input to stage II.

The proposed density-based SO exploits a property of the filtering step to turn a standard TO into a SO: Using the TO methods from section 2 in stage I, the interface of the obtained structure is not sharp but blurred out due to the averaging effect of the density filter (cf.

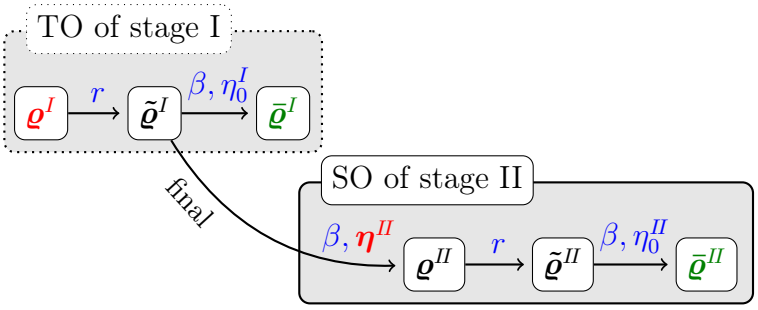


Figure 5.1 – The basic outline of density-based shape optimization: Two optimization stages with their respective DVs, displayed in red.

equation (2.2)). The projection step of equation (2.3) is thus applied to make the gradual interface region more compact for increasingly larger β values such that the interface appears sharper (see Appendix C.2). Nevertheless, when modifying the projection threshold value η, the outline of the part can be altered.

This effect is used for example to include the effects of uniform (see section 2.6.1) or spatially varying (see [11, 86, 109]) manufacturing errors in a TO, in which eroded and dilated designs are calculated besides the nominal design by altering η values. Appendix C.3 demonstrates that a uniform variation of the projection threshold η shifts the outline of the part perpendicular to the spatial gradient of the filtered field.

In contrast to that, for the presented SO method, possible shifts of the part's outline are not prescribed but instead are to be determined by the SO, which represents stage II of the overall optimization. Hence, a field of $\boldsymbol{\eta}^{II}$ values is used in optimization stage II to apply different projection thresholds to each finite element in the design space.

The differences between the two optimization stages may thus be summarized as follows:

- stage I: $\boldsymbol{\varrho}^I$ can directly be altered by the optimizer.
- stage II: $\boldsymbol{\varrho}^{II}$ is the result of $\tilde{\boldsymbol{\varrho}}^I$ projected using $\boldsymbol{\eta}^{II}$. $\boldsymbol{\eta}^{II}$ is now the design vector and is altered by the optimizer.

The remaining further variable processing steps of the SO in stage II are adapted from the three field TO approach.

However, it has to be kept in mind that a SO generally offers less design freedom than a TO, since it cannot generate new topologies. The proposed two-stage optimization procedure therefore can only be advantageous, if the objective, constraints, or other modeling aspects change in between the optimization stages and the SO stage converges faster than a new TO would do if restarted from scratch.

In the following, the individual variable processing steps for the SO of stage II shown in figure 5.1 are discussed as well as the setup of the corresponding optimization problem.

5.1.1 Design variables of stage II

In stage II, the $\boldsymbol{\eta}^{II}$ values are the DVs. Each entry η_i^{II} defines an own projection threshold for every finite element i of the model.

The projection equation (2.3) is used to transform the final filtered variables $\tilde{\boldsymbol{\varrho}}^I$ of the

preliminary design from optimization stage I to the intermediate variables $\boldsymbol{\varrho}^{II}$ (cf. figure 5.1):

$$\boldsymbol{\varrho}^{II} = \boldsymbol{\varrho}^{II}(\tilde{\boldsymbol{\varrho}}^{I}, \boldsymbol{\eta}^{II}, \beta), \qquad \tilde{\boldsymbol{\varrho}}^{I} = \text{const}, \quad \beta = \text{const} \tag{5.1a}$$

$$\eta_i^{II} \in [0,1], \qquad i = 1, \dots n_e \tag{5.1b}$$

The filtered variables $\tilde{\boldsymbol{\varrho}}^{I}$ and the parameter β are taken from the final iteration of the converged TO from stage I and are therefore constant.

5.1.2 Filtering and final projection in stage II

Using element-wise projection thresholds η_i^{II} allows altering the entries of $\boldsymbol{\varrho}^{II}$ for every element independently. The effect of the density filter used in stage I is thus nullified. As a result, mesh independence is lost and numerical artifacts like checkerboarding may appear again. To counter this, another filtering and projection step is done in stage II, e.g. with a constant uniform threshold $\eta_0^{II} = 0.5$ (cf. figure 5.1):

$$\tilde{\boldsymbol{\varrho}}^{II} = \tilde{\boldsymbol{\varrho}}^{II}(\boldsymbol{\varrho}^{II}, r) \tag{5.2}$$

$$\bar{\boldsymbol{\varrho}}^{II} = \bar{\boldsymbol{\varrho}}^{II}(\tilde{\boldsymbol{\varrho}}^{II}, \eta_0^{II}, \beta) \tag{5.3}$$

The further variable processing in stage II therefore follows the standard three field TO approach presented in section 2.3. As a result, also all previously mentioned extensions to a TO can be directly incorporated into a density-based SO, as discussed later in section 5.1.6.

5.1.3 Optimization problem formulation

The basic optimization problem for a density-based SO, with an objective function Φ and constraint functions h_i, states (cf. equations (2.1)):

$$\min_{\boldsymbol{\eta}^{II}} \Phi(\boldsymbol{\eta}^{II}) \tag{5.4a}$$

$$\text{s.t.} \quad h_i(\boldsymbol{\eta}^{II}) \leq 0, \quad \forall i \tag{5.4b}$$

$$0 \leq \eta_j^{II} \leq 1, \quad j = 1, \dots, n_e \tag{5.4c}$$

$$\boldsymbol{K}(\boldsymbol{\eta}^{II})\boldsymbol{u} = \boldsymbol{f} \tag{5.4d}$$

5.1.4 Sensitivities in stage II

The presented method for the density-based SO requires minimal adaptation of already existing TO codes following the three field approach. Only a projection step has to precede the standard three field processing. Therefore, gradients obtained by a TO code have to be multiplied by another partial derivative to account for this projection step, in order to obtain the gradients for the SO:

$$\frac{d(\cdot)}{d\eta_i^{II}} = \frac{d(\cdot)}{d\varrho_i^{II}} \frac{d\varrho_i^{II}}{d\eta_i^{II}} \tag{5.5}$$

The partial derivative $\frac{d\varrho_i^{II}}{d\eta_i^{II}}$ is the gradient of the projection function with respect to the threshold variable and is given in equation (D.9) in Appendix D.4.

5.1.5 Initial values in stage II

The SO design variables $\boldsymbol{\eta}^{II}$ are mostly initialized with the value η_0^I from stage I for each entry. Consequently, for the initial state (iteration 0) of stage II, the following equalities hold:

$$\boldsymbol{\eta}^{II}_{initial} = [\eta_0^I, \ldots, \eta_0^I]^T \tag{5.6a}$$

$$\boldsymbol{\varrho}^{II}_{initial} = \bar{\boldsymbol{\varrho}}^I_{final} \tag{5.6b}$$

Since an additional filtering and projection step is employed in stage II, the resulting projected variables are very close to each other but not exactly equal:

$$\bar{\boldsymbol{\varrho}}^{II}_{initial} \neq \bar{\boldsymbol{\varrho}}^I_{final} \tag{5.7}$$

While the SO of stage II is a separate optimization, it takes filtered variables from a preceding TO as input. Therefore, effectively, a double filtering and projection is applied[13]. For the studied examples, the deviations introduced by double filtering when compared to single filtering are negligible and the optimization is dominated by the choice of the design variables in $\boldsymbol{\eta}^{II}$. In addition, it is not necessary for the initial design of stage II to exactly reproduce the preliminary design, since the goal of the SO is to deviate from the preliminary shape in order to obtain an optimized design.

5.1.6 Combination with other topology optimization methods

Since the presented equations for a density-based SO can be wrapped around existing TO codes following the three field approach, all possible modifications of a plain TO discussed in sections 2.6, 4, and 6 can also be applied within the proposed SO method.

To name a few possible combinations, e.g. methods for size control (see section 2.6.3) involving local volume constraints for controlling the maximum member size or the "robust" approach to obtain designs with a minimum feature size can be employed within the SO of stage II. Also, methods for including partial failure during the optimization (see section 4) can be used together with the proposed SO method, which is the main topic of the following sections 5.2 – 5.5.

5.2 Application to fail-safe optimization

To drastically reduce the computational effort required for an optimization towards a fail-safe design, a load-path-based damage model is presented in section 4.3. However, tackling a load-path-based fail-safe optimization via TO methods leads to convergence issues like e.g. a "switching" of topologies, as discussed in section 4.3.5. The reason for this is the unsteady change of the underlying optimization problem.

To remedy this issue, the idea is to alter only the shape of a part in a SO and keep its topology constant. By using a topology preserving optimization method, also the identified load paths of the studied structure do not need to be updated once they are identified at the beginning, such that also the optimization problem does not change.

The two-stage optimization procedure discussed in this section, involving a density-based SO, is therefore suggested as a means to optimize for fail-safe designs within a reasonable amount of computational time, as presented by the author in [58, 101].

[13] Double filtering within a single optimization is successfully applied e.g. in [79] for the "robust" TO approach for highly sensitive optimization problems, where a single filtering step proved to be insufficient.

As discussed before, stage I is a TO that generates a preliminary design. However, stage I does not yet include damage considerations and comes therefore at a low computational cost. In between the optimization stages, the load path identification from section 4.3.4 is performed once to generate a set of n_d damage scenarios, each affecting a single load-bearing member. In the SO of stage II, the outline of the structure is optimized, this time including damage considerations on the (constant) set of load paths.

The workflow of the proposed two-stage fail-safe optimization procedure is shown in figure 5.2. As shown in section 4.2.1, designs obtained by a redundancy enforcing TO have the potential to offer increased damage tolerance, however, they are not optimized in this matter. Nevertheless, such designs are good candidates for a preliminary design and can be obtained cheaply. The cost-intensive part is the SO of stage II, since in every iteration the n_d damage scenarios need to be simulated.

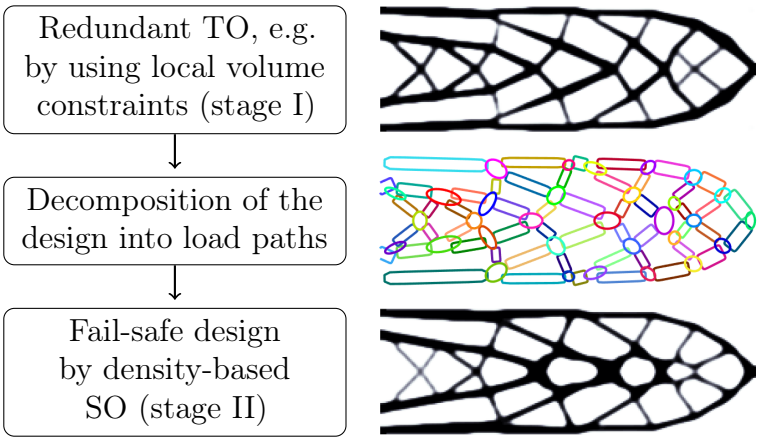


Figure 5.2 – Workflow for the proposed two-stage fail-safe optimization (cf. [58]).

The two-stage procedure from figure 5.2 offers several advantages compared to the TO methods for fail-safe optimization from section 4.2 using the simplified rectangular damage model from [35]: At first, far fewer damage cases need to be considered, if a load-path-based damage model is used. Also, damage cases are only considered during the SO, that is at a stage, where the part's geometry is already developed, unlike for pure TO methods, where damage scenarios are also calculated for all intermediate results which still may have large unconverged areas. In this matter, the two-stage procedure exhausts the idea behind the threshold method from section 4.2.3. Additionally, rapid convergence of the SO stage is observed, as shown later in the examples in section 5.3. Therefore, far fewer iterations involving the computationally expensive damage considerations are needed compared to a fail-safe TO starting from scratch. Finally, using a two-stage procedure with a fast stage I allows to visually inspect the topology of the design already at a very early "draft" stage and not only after a long fail-safe optimization. This inspection could relate to whether e.g. the count and distribution of redundant load paths meet the expectations of the user or if modeling parameters need to be adapted.

The simplified variable processing diagram for the proposed two-stage optimization for fail-safe design is shown in figure 5.3.

For stage I, two cases will be studied in the examples in the following section 5.3: Either a TO with the "robust" approach is performed to obtain a preliminary design with a simple topology, which only serves as a reference, or a TO with enforced redundancy by local volume constraints is performed.

Stage II combines the basic SO method presented in figure 5.1 with the parametric load-path-

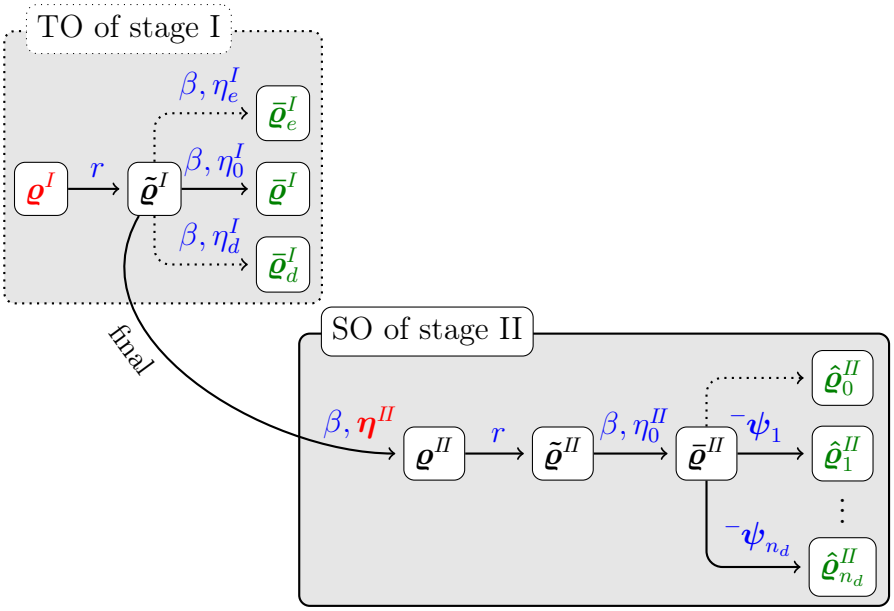


Figure 5.3 – Variables in a density-based shape optimization for fail-safe design.

based damage model from section 4.3, involving mask vectors $\bar{\boldsymbol{\psi}}_d$ that define the n_d individual damage scenarios and yield the modified densities $\hat{\boldsymbol{\varrho}}_d^{II}$ as fourth density fields.

5.2.1 Definition of the damage scenarios

The load path identification method mentioned in section 4.3.4 and Appendix B.1 is used to automatically define the damage scenarios and derive the corresponding damage masks $\bar{\boldsymbol{\psi}}_d$ that are considered in stage II. The identification is only done once on the final design of stage I. Since this is the baseline design for stage II and the SO therein only alters the outline of the part, the basic decomposition of the part into its load paths stays the same throughout the optimization, which is not true for the method involving a TO discussed in section 4.3.5.

For the SO in stage II, special care has to be taken regarding the scale of the damage shapes, from which the damage masks are calculated: The damage shapes need to fully enclose the respective load-bearing member, even if its outline is slightly altered throughout the SO, requiring some excess size to compensate for this movement. In this work, a simple up-scaling to twice the minimal size is applied, as already discussed in section 4.3.4. The damage shapes for beams are additionally altered to compact cuts, which then yield damage shapes like the ones shown in figure 4.14b and 4.14d.

In contrast to the simplified rectangular damage model discussed in section 4.2, the load-path-based damage model is robust with relation to slight changes of the size and position of the damage shapes (cf. Appendix B.2), as long, as each damage shape still mostly affects only the designated load path. However, the sizes of the damage shapes should not be chosen too small, since then a circumvention effect is facilitated. If this effect is exploited by the optimizer, the damage modeling becomes compromised. In contrast to that, larger-than-required damage shapes seem to have no negative effect on the fail-safe optimization, at least for compliance as objective. Both just mentioned aspects are demonstrated in Appendix B.4.

5.2.2 Minimizing the maximum objective value for all damage scenarios

Following the approach from section 4.3, each of the n_d considered damage scenarios is described by a vector of modified densities $\hat{\boldsymbol{\varrho}}_d^{II}$ that is obtained from the projected densities $\bar{\boldsymbol{\varrho}}^{II}$ and the corresponding damage mask vector ${}^{-}\boldsymbol{\psi}_d$ (cf. figure 5.3). Evaluating the structural response for each damage scenario yields the corresponding objective value Φ_d:

$$\Phi_d := \Phi(\hat{\boldsymbol{\varrho}}_d^{II}) = \Phi\left(\hat{\boldsymbol{\varrho}}_d(\boldsymbol{\eta}^{II}, {}^{-}\boldsymbol{\psi}_d)\right), \quad d = \{0, 1, \ldots, n_d\} \tag{5.8}$$

The objective Φ_d with the maximum value over all considered damage scenarios shall be minimized in stage II. Compared to equation (4.5), covering the same problem for TO, the corresponding min-max problem now depends on the shaping DVs $\boldsymbol{\eta}^{II}$:

$$\min_{\boldsymbol{\eta}^{II}} \max_{d} \Phi_d(\boldsymbol{\eta}^{II}), \quad d = \{0, 1, \ldots, n_d\} \tag{5.9}$$

Equation (5.9) is approximated by the KS aggregation function (A_{KS} from section 3.2.3, offering a parameter p), such that a differentiable expression is obtained (cf. equation (4.6)):

$$\min_{\boldsymbol{\eta}^{II}} \frac{1}{p} \log\left(\sum_{d=0}^{n_d} \mathrm{e}^{p\Phi_d(\boldsymbol{\eta}^{II})}\right) \tag{5.10}$$

The overall constrained optimization problem for a worst-case fail-safe SO task is obtained by using equations (5.4) and exchanging (5.4a) by (5.10).

For the studied cases in this work, where compliance is the objective function, the undamaged case ($d = 0$) does not have to be considered for the optimization (see section 4.1.4), thus one FE simulation is saved per optimization iteration. The worst-case compliance optimization task then states:

$$\min_{\boldsymbol{\eta}^{II}} \frac{1}{p} \log\left(\sum_{d=1}^{n_d} \mathrm{e}^{pc_d(\boldsymbol{\eta}^{II})}\right) \tag{5.11a}$$

$$\text{s.t.} \quad h_i(\boldsymbol{\eta}^{II}) \leq 0, \quad \forall i \tag{5.11b}$$

$$0 \leq \eta_j^{II} \leq 1, \quad j = 1, \ldots, n_e \tag{5.11c}$$

$$\boldsymbol{K}(\boldsymbol{\eta}^{II})\boldsymbol{u} = \boldsymbol{f} \tag{5.11d}$$

5.2.3 Volume constraint

In all examples shown in this section, the only constraint h used in the SO of stage II is the global volume constraint from section 2.4.2. In order to make the results of stages I and II comparable, and to be able to judge the improvements by the SO, the global volume constraint is used to prohibit that the total volume of design from stage II exceeds the volume of the design from stage I. Therefore, the global volume fraction ω is limited to a maximum value k_g, which is determined by the final design from the TO of stage I:

$$h = \omega(\boldsymbol{\eta}^{II}) - k_g \leq 0 \tag{5.12}$$

The part's volume fraction ω is evaluated for the intact, undamaged configuration.

Other constraints, like e.g. constraints imposing feature size restrictions, are not used in stage II, such that the proposed SO method can unfold its full potential.

5.2.4 Sensitivities

For the fail-safe optimization in stage II, partial derivatives with respect to the undamaged and the damaged configurations are utilized. Ultimately, the total derivatives with respect to the shaping DVs $\boldsymbol{\eta}^{II}$ are needed.

In combining the information from section 4.1.3 about how to link sensitivities from damaged and undamaged configurations in a TO and from section 5.1.4, dealing with augmenting TO to SO sensitivities, all required expressions can be obtained. However, for convenience, the required total derivatives are summarized in the following:

The volume constraint h from equation (5.12) is calculated on the undamaged configuration, the total derivative is therefore[14]:

$$\frac{dh}{d\eta_i^{II}} = \sum_j \frac{\partial h}{\partial \bar{\varrho}_j^{II}} \frac{\partial \bar{\varrho}_j^{II}}{\partial \tilde{\varrho}_j^{II}} \frac{\partial \tilde{\varrho}_j^{II}}{\partial \varrho_i^{II}} \frac{\partial \varrho_i^{II}}{\partial \eta_i^{II}} \tag{5.13}$$

The objective function is evaluated on the damaged configurations. The total derivative, involving the damage mask ${}^{-}\boldsymbol{\psi}_d$, is thus[15]:

$$\frac{d\Phi_d}{d\eta_i^{II}} = \sum_j \frac{\partial \Phi_d}{\partial \hat{\varrho}_{d,j}^{II}} \frac{\partial \hat{\varrho}_{d,j}^{II}}{\partial \bar{\varrho}_j^{II}} \frac{\partial \bar{\varrho}_j^{II}}{\partial \tilde{\varrho}_j^{II}} \frac{\partial \tilde{\varrho}_j^{II}}{\partial \varrho_i^{II}} \frac{\partial \varrho_i^{II}}{\partial \eta_i^{II}} \tag{5.14}$$

$$= \sum_j \frac{\partial \Phi_d}{\partial \hat{\varrho}_{d,j}^{II}} \frac{\partial \bar{\varrho}_j^{II}}{\partial \tilde{\varrho}_j^{II}} \frac{\partial \tilde{\varrho}_j^{II}}{\partial \varrho_i^{II}} \frac{\partial \varrho_i^{II}}{\partial \eta_i^{II}} {}^{-}\psi_{d,j} \tag{5.15}$$

5.2.5 Alternative formulation of the fail-safe optimization problem

The fail-safe optimizations discussed in this work adopt the optimization formulation from [35], where the worst-case objective function under damage is minimized while e.g. a global volume constraint is employed (cf. equations (5.11) and (4.7)). However, strictly speaking, this is an optimization formulation aiming at increasing the damage tolerance of the studied structure but it does not guarantee, that the worst-case objective value stays within certain limits.

Nevertheless, for some applications, it may be desirable to swap the objective and the constraint function in the optimization problem. In the context of this work, this would mean that an optimization problem is set up to e.g. minimize the global volume fraction ω while limiting the worst-case compliance to an upper threshold value $\bar{c}_t$, which will be referred to as alternative formulation for a fail-safe optimization in the following. The optimization problem then states:

$$\min_{\boldsymbol{\eta}^{II}} \omega(\boldsymbol{\eta}^{II}) \tag{5.16a}$$

$$\text{s.t.} \quad \frac{1}{p} \log\left(\sum_{d=0}^{n_d} \mathrm{e}^{p\Phi_d(\boldsymbol{\eta}^{II})} \right) \leq \bar{c}_t \tag{5.16b}$$

$$0 \leq \eta_j^{II} \leq 1, \quad j = 1, \ldots, n_e \tag{5.16c}$$

$$\boldsymbol{K}(\boldsymbol{\eta}^{II})\boldsymbol{u} = \boldsymbol{f} \tag{5.16d}$$

[14] The four partial derivatives listed are found in their given order in equations (2.14), (D.8), (D.7), and (D.9).
[15] The four partial derivatives of the end result are given in section 2.4 and equations (D.8), (D.7), and (D.9).

The goal of the optimization is thus to find the lightest part, that does not exceed a certain displacement at the loading point, even when damage on single load-bearing members occurs.

This type of fail-safe optimization problem is challenging, since, depending on the desired threshold $\bar{c}_t$ for the worst-case compliance, a feasible solution cannot be guaranteed within the limits posed by the constraints of the optimization. Even if a feasible solution exists, the initial design will in most cases not yet fulfill the desired worst-case performance, such that the optimization at least starts in the infeasible domain.

5.3 Numerical examples for fail-safe design considering damage

Examples for the fail-safe optimization of 2D and lattice-type 3D problems are given in this section. The structures are optimized for a minimal worst-case compliance under damaged conditions via the proposed two-stage SO procedure.

For all shown examples, the SO in stage II converges very fast and only 25 iterations considering damage scenarios are conducted. The corresponding convergence plots are given in Appendix B.5 and underline, that the majority of the improvement in objective values is achieved even within the first 10 iterations.

For the 2D case, the two examples given in section 2.7 are considered: A cantilever beam and an L-shaped bracket. For both problems, at first, the design with the simplest obtained topology is studied. Since these designs do not offer redundant load paths, the effect of a failure of a load-bearing member is expected to be severe. For that reason, these designs are not recommended as preliminary designs for a SO, if aiming for a design with high damage tolerance. Nevertheless, due to their simple topology, these designs serve as easy-to-understand demonstrations of the proposed method. Afterward, the corresponding redundant designs are used as preliminary designs. Due to the redundant load paths present, these base designs are potentially more robust towards damage from the start on.

For 3D problems, a plain TO or a TO with the "robust" approach yield plate-like structures, if the FE discretization is fine enough. However, this work focuses on lattice-type structures, since for these structures the automated method for the definition of damage scenarios on load-bearing members presented in section 4.3 is applicable. Therefore, local volume constraints are used to break down plates into lattice grids in the 3D examples. While lattice structures are not optimal for the undamaged case, they offer some advantages when it comes to damage tolerance in real-world applications: Structural damage induced by a fracture may grow and thus affect a whole plate-like part but will stop in a lattice-type part, after the affected lattice member is fully broken.

5.3.1 Fail-safe minimum compliance 2D cantilever design

The design space, loading, and BCs for the fail-safe minimum compliance cantilever problem are shown in figure 5.4. Again, a region at the tip of the cantilever is excluded from damage considerations, i.e. no damage will be applied in regions right of the blue line shown in figure 5.4. This ensures, that the point of load application is never fully disjoined from the remaining structure by a single damage case.

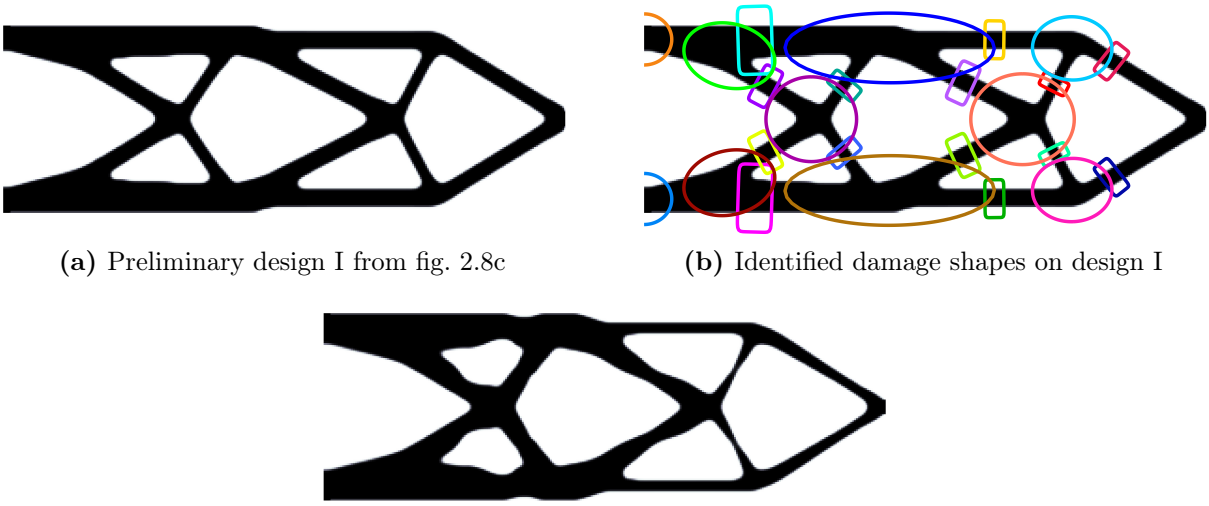
Non-redundant design The design shown in section 2.7.1 obtained by the "robust" approach is used as an example for a non-redundant design and is repeated in figure 5.5a. The outlines of the identified damage shapes (14 beams and 10 knots) for this design are shown in figure 5.5b.

The weakness of this design is, that the part is connected at only 2 locations to the fixed boundary on the left. The worst-case damage for the starting design is therefore the failure

Figure 5.4 – Design space with BCs, loads, and border for the damage region (blue) for the fail-safe 2D cantilever example.

of a knot close to one of the two load paths connecting the structure to the fixed boundary. While the compliance of the undamaged structure has a relatively low value of $c_0 = 213.68$, the compliance drastically increases to a value of $\bar{c} = 12\,308.41$ for this worst-case damage scenario.

By using density-based SO, the performance under failure conditions can be improved within a few optimization steps considering all 24 damage cases. Fast convergence is observed and after 25 iterations the result from figure 5.5c is obtained. The SO stage conserves the topology and also the volume of the part is unchanged. The worst-case compliance is significantly reduced by 40% to a value of 7 364.49. This reduction comes at the price of an increased compliance of less than 8% for the nominal, undamaged case.



(a) Preliminary design I from fig. 2.8c

(b) Identified damage shapes on design I

(c) Shape optimized design II

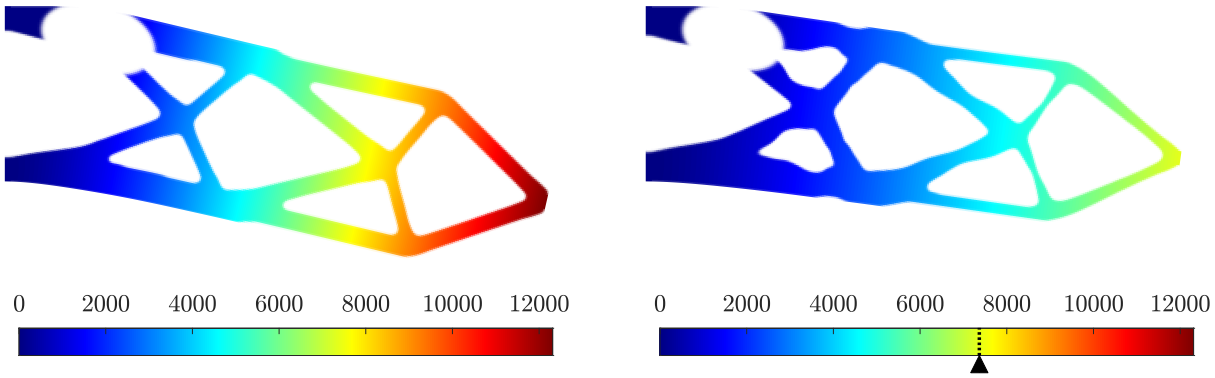
Figure 5.5 – Initial and shape optimized design for the 2D cantilever beam example.

Table 5.1 summarizes the numerical values for the worst-case compliance $\bar{c}$, undamaged compliance c_0, and the global volume fraction ω for the initial and the shape optimized design.

Plots of the absolute transversal displacements for both designs under their respective worst-case damages are shown in figure 5.6.

Table 5.1 – Compliance values and volume fractions for the shaped 2D cantilever beam.

Design	Figure	$\bar{c}$	c_0	ω
Design I	5.5a	12 308.41	213.68	40.02%
Design II	5.5c	7 364.49	230.26	40.02%



(a) Initial design I **(b)** Shape optimized design II

Figure 5.6 – Absolute transversal displacements for design I and II under their worst-case damage.

The preliminary design I consists of beams with mostly constant cross-sections (cf. fig. 5.5a), since the structure is dominated by tension and compression section forces. In contrast to this, the beams of the shape optimized design from figure 5.5c are altered into beams of varying widths, which means the sections are dominated by bending moments. The distribution of the bending moments in the case of partial failure is analyzed in Appendix B.3 and proves, that under damaged conditions the inner loading is redistributed such, that in many beams the function of the moment is zero in the center. To increase the bending resistance, the moments of inertia are optimized for this loading and, consequently, beams with thin middle sections are obtained in stage II.

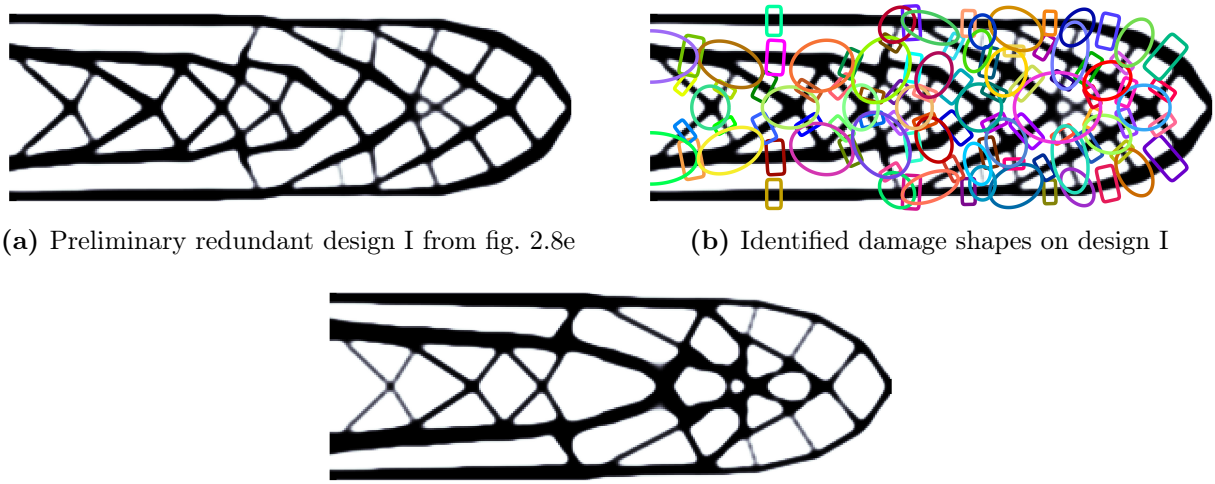
Redundant design Using the design with enforced redundancy from section 2.7.1 as preliminary design I has the advantage, that this design remedies the weakness of the non-redundant design in having four distinct connections to the fixed boundary on the left. Together with the overall much larger number of beams and connections, this part has the potential to be less sensitive to the failure of single members, since then alternative load paths are close by. The increased robustness of structures obtained using local volume constraints towards local failure is discussed e.g. in [44].

Figure 5.7 shows the starting design, the outlines of the damage shapes, and the shape optimized design after 25 iterations for the redundant cantilever. 64 beams and 37 knots are identified and their respective 101 damage shapes are considered for the fail-safe optimization.

Since the preliminary design in figure 5.7a offers some very thin beams, the SO stage is in this case able to let some less important beams disappear, such that a simpler topology is obtained in figure 5.7c.

The compliance for the worst-case damage scenario is $\bar{c} = 1\,146.21$ and therefore one order of magnitude better than for the non-redundant example discussed earlier. Nevertheless, by using the proposed SO method, the worst-case compliance is further cut down and is halved to a value of 576.61. In this case, even the nominal compliance for the undamaged case was improved by about 4% during stage II. This is due to the less restrictive global volume constraint used in stage II, allowing a more flexible redistribution of material compared to the local volume constraint used in stage I, where it was used only as a means to obtain a redundant design in the first place.

The nominal compliances c_0 of the redundant cantilever beams are clearly worse, than for their non-redundant counterparts. However, it also has to be considered, that the volume



(a) Preliminary redundant design I from fig. 2.8e

(b) Identified damage shapes on design I

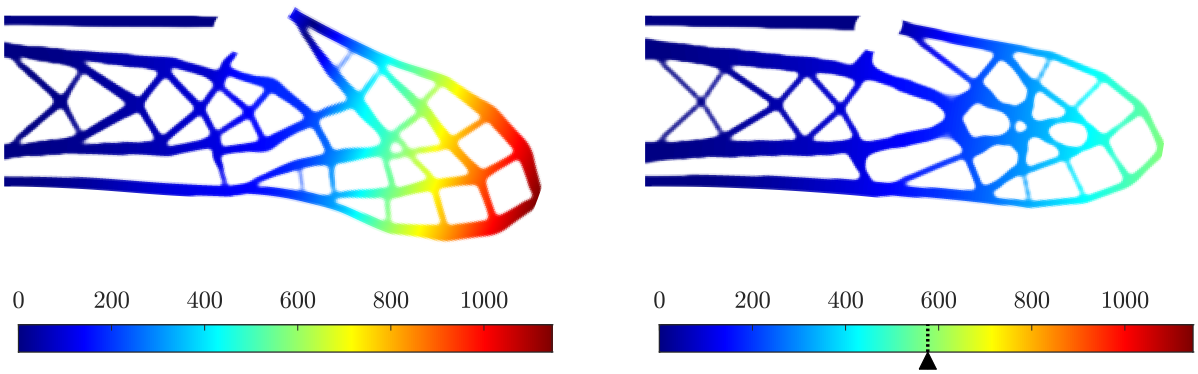
(c) Shape optimized design II

Figure 5.7 – Initial and shape optimized design for the redundant 2D cantilever beam example.

fractions are lower. Table 5.2 lists the worst-case and nominal compliance values ($\bar{c}$ and c_0, respectively), together with the global volume fractions ω of both designs. The absolute transversal displacements for the worst-case damage scenarios are shown in figure 5.8.

Table 5.2 – Compliance values and volume fractions for the shaped redundant 2D cantilever beam.

Design	Figure	$\bar{c}$	c_0	ω
Design I	5.7a	1 146.21	328.30	37.17%
Design II	5.7c	576.61	315.59	37.17%



(a) Initial design I

(b) Shape optimized design II

Figure 5.8 – Absolute transversal displacements for design I and II under their worst-case damage.

5.3.2 Fail-safe minimum compliance 2D L-bracket design

Figure 5.9 shows the design space, BCs, and the load for the fail-safe L-bracket problem. A blue line marks the region, where no damage is applied in order to prevent the loaded point from being cut away directly.

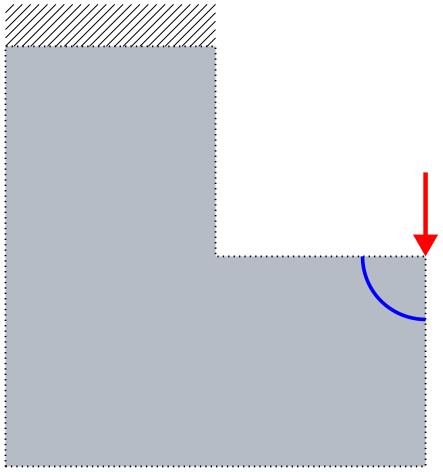
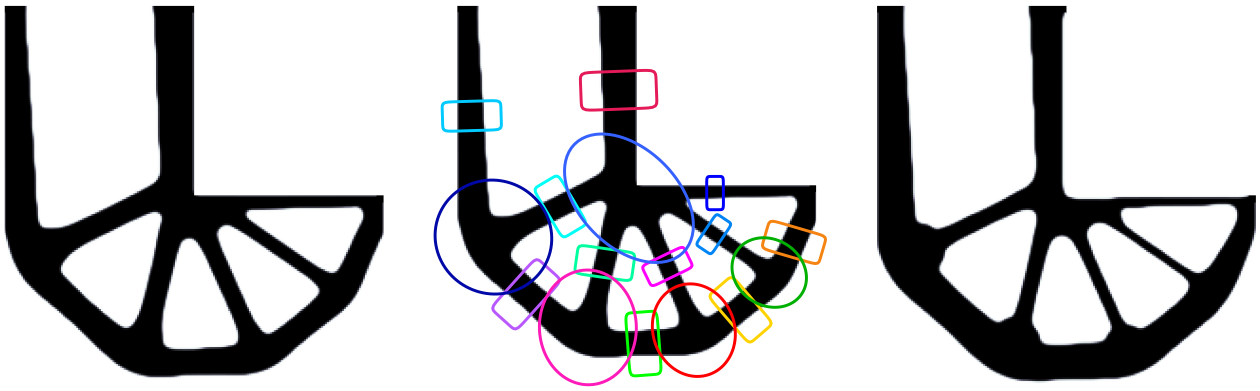


Figure 5.9 – Design space with BCs, loads, and border for the damage region (blue) for the fail-safe 2D L-bracket example.

Non-redundant design The L-bracket design from section 2.7.2 obtained by the "robust" approach is used as a non-redundant example for a preliminary design I for the proposed method. The design is repeated in figure 5.10a. Figure 5.10b shows the outlines of the damage shapes for the 16 identified structural members (11 beams and 5 knots).

Most of the beams of the structure extend radially from the inner corner of the L-bracket. Therefore, the material around the inner corner is identified as a knot region and this knot becomes the critical weakness of the design. If the structure fails at this knot, all load has to go through the outer arc and the left vertical beam. The compliance in this case drastically increases from a value of $c_0 = 100.97$ for the undamaged case to a value of $\bar{c} = 28\,988.87$.

Since a damage at the inner corner has such a severe impact, it dominates the whole optimization in stage II. The SO therefore transfers as much material to the outer arc and left vertical beam as possible and thins all other members, such that the result from figure 5.10c is obtained. The effect of the damage is again considerably reduced by 44% to a worst-case compliance value of 16 198.23, which however stays at a very high level when compared to the nominal value for the undamaged case. This again underlines, that a non-redundant preliminary design in stage I is a suboptimal choice if aiming for high damage tolerance.



(a) Preliminary design I from fig. 2.9c

(b) Identified damage shapes on design I

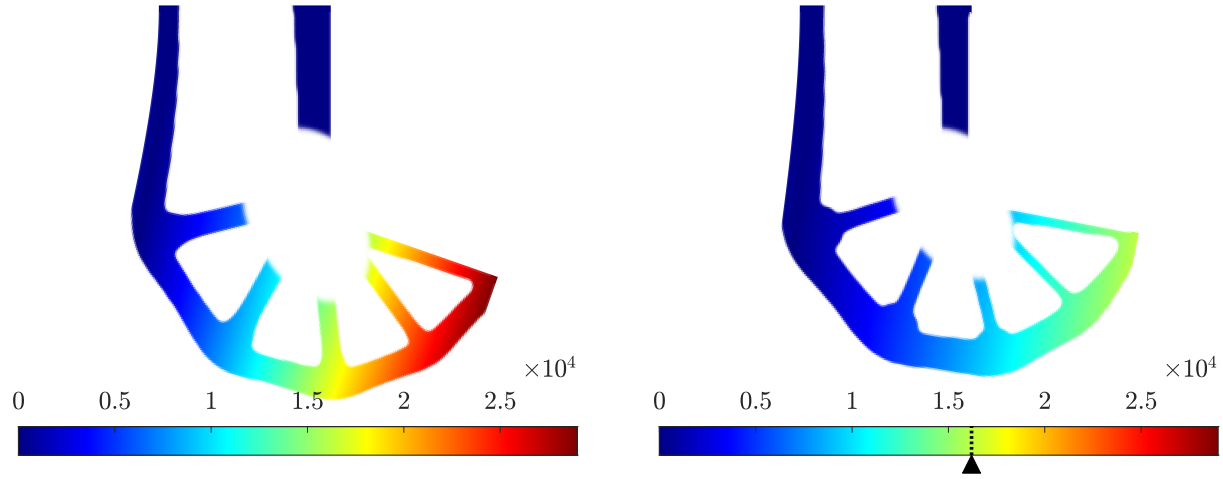
(c) Shape optimized design II

Figure 5.10 – Initial and shape optimized design for the 2D L-bracket example.

Table 5.3 lists the compliance values $\bar{c}$ and c_0 for the worst-case damage scenario and undamaged case, respectively, along with the global volume fractions ω of the designs before and after the SO stage II. The absolute vertical displacements for the worst-case damage at the inner corner are shown in figure 5.11.

Table 5.3 – Compliance values and volume fractions for the shaped 2D L-bracket.

Design	Figure	$\bar{c}$	c_0	ω
Design I	5.10a	28 988.87	100.97	40.02%
Design II	5.10c	16 198.23	110.67	40.02%



(a) Initial design I

(b) Shape optimized design II

Figure 5.11 – Absolute vertical displacements for design I and II under their worst-case damage.

Redundant design Figure 5.12a shows the L-bracket design from section 2.7.2 that is obtained using local volume constraints on circular test regions. Thick beams are clearly split into thinner beams when compared to the non-redundant design from the previous paragraph. Critical regions, like e.g. the vertical beams connecting the structure to the fixed boundary at the top or the whole outer arc of the bracket offer redundant load paths. However, this design still shares the main vulnerability with the non-redundant design: A damage at the inner corner would again require the load to go through the outer arc. Although the arc now consists of a pair of concentric beams, it would still be subjected to very high inner loads.

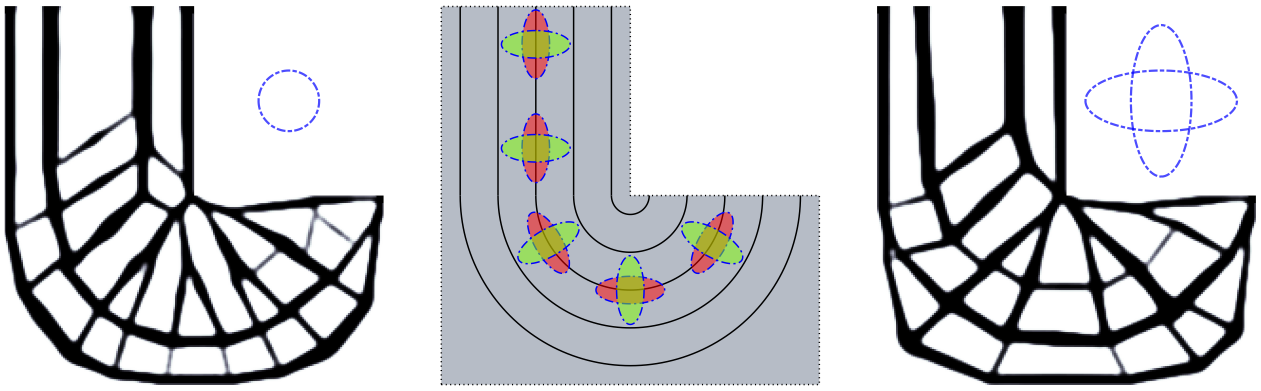
The main cause for the persistence of the above-mentioned weakness is, that the design from figure 5.12a still has some very long beams radially extending from the region around the inner corner. In between these beams, large areas of void exist. Additional cross-members in these void regions would significantly improve the robustness towards a failure in the corner region. However, unlike for the cantilever beam example discussed in section 5.3.1, the simple approach of using local volume constraints with circular test regions of the chosen size is not sufficient to obtain a relatively regular grid of cross-members for the L-bracket.

One solution to cope with this problem is to reduce the size of the local volume constraint test regions. This would increase the overall number of members and the design would approach

a "porous" design (see [44]). However, in this case, also the computational cost for the proposed fail-safe optimization is increased, since then more failure cases are considered in the optimization.

Another approach is to use the anisotropic filter, as proposed in the original paper [44]. Therein, orthogonal elliptic test regions are recommended for certain optimization problems, instead of circular regions. As outlined in [44], elliptic test regions help to overcome the problem of material being strictly aligned in one direction and having only sparsely connected orthogonal directions.

This work follows the latter approach and uses ellipses with a semi-minor axis corresponding to the original radius of the circular test region (see figure 5.12a, the radius is 16 or four times the filter radius). The value for the semi-minor axis controls the overall degree of redundancy (or porosity), just like the radius does for the circular test regions. The semi-major axis is responsible for the desired anisotropic effect. For aspect ratios close to 1, the ellipses approach the circular test region and no anisotropic effect is visible. In this work, the semi-major axis is chosen as 2.5 times the length of the semi-minor axis. In [44] it is proposed to align the ellipses with the principal stress directions obtained by an FE analysis of the initial design space. However, in this case, since the design space has a simple geometry, a yet simpler approach is sufficient, which does not require an additional calculation: The ellipses are aligned with relation to the medial axis of the design space, as shown in figure 5.12b. The actual size of the ellipses together with the obtained alternative redundant L-bracket design are shown in figure 5.12c.



(a) Design with circular test region from figure 2.9e

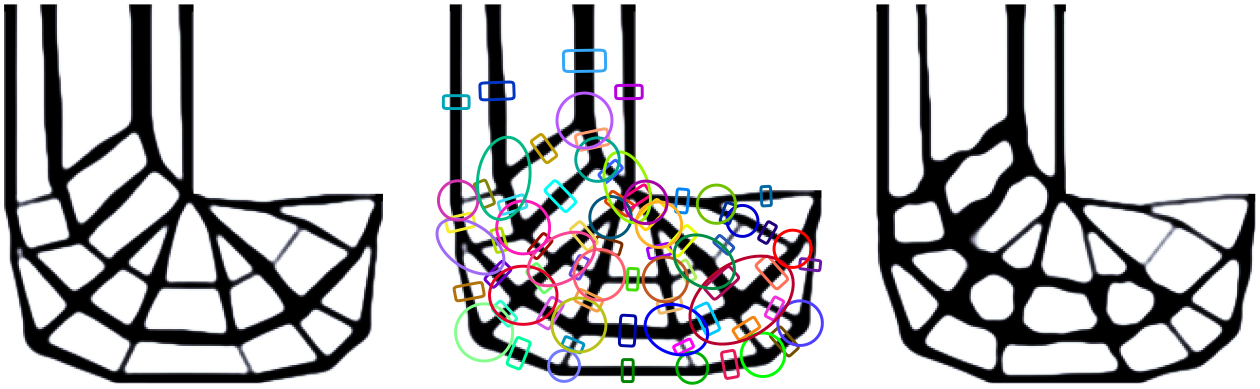
(b) Anisotropic test region orientations

(c) Design with elliptic test regions

Figure 5.12 – Effect of different test regions of the local volume constraints on the redundant L-bracket design.

Due to the additional cross-members near the inner corner, the design from figure 5.12c offers increased robustness towards failure and is a good preliminary design for the proposed SO method. In total, 74 structural members are identified (48 beams and 26 knots). The corresponding damage shapes and the shape optimized design are shown in figure 5.13.

The initial redundant design has a worst-case compliance of $\bar{c} = 2\,713.01$ and is in this matter again one order of magnitude better, than the previously discussed non-redundant L-bracket. The SO in stage II further decreased the worst-case compliance by 68% to a value of 859.03 within 25 iterations. Again, the redundant designs have a worse compliance c_0 for the undamaged case, compared to the non-redundant designs, but also the volume fractions are lower. Table 5.4 lists the worst-case and nominal compliance values, and the volume fractions. Figure 5.14 shows the absolute vertical displacements of both designs for their respective worst-case damage at the inner corner.

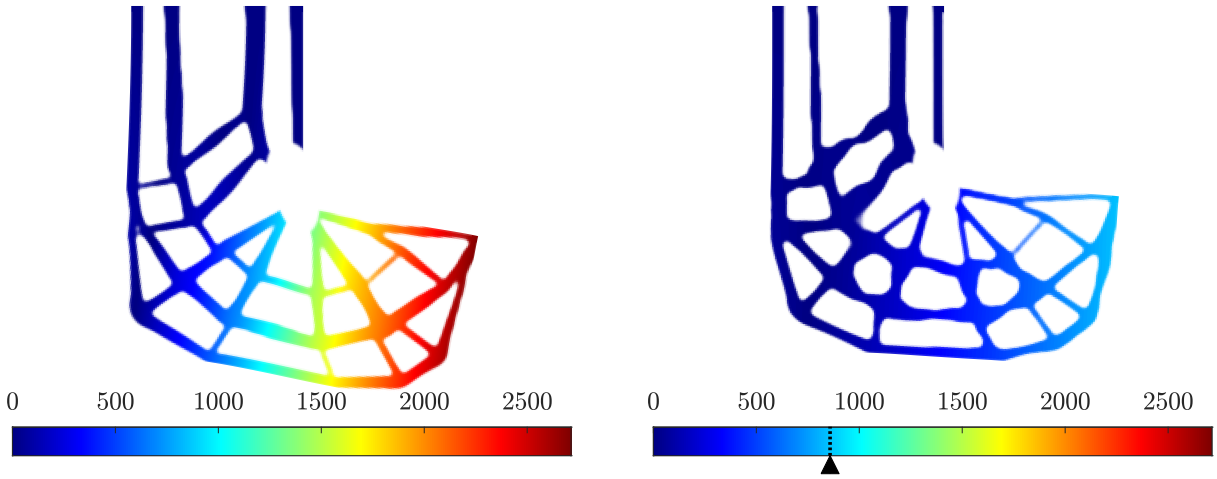


(a) Preliminary redundant design I **(b)** Identified damage shapes on design I **(c)** Shape optimized design II

Figure 5.13 – Shape optimized design for the alternative redundant 2D L-bracket example.

Table 5.4 – Compliance values and volume fractions for the shaped redundant 2D L-bracket.

Design	Figure	$\bar{c}$	c_0	ω
Design I	5.13a	2 713.01	144.05	37.39%
Design II	5.13c	859.03	163.29	37.38%



(a) Initial design I **(b)** Shape optimized design II

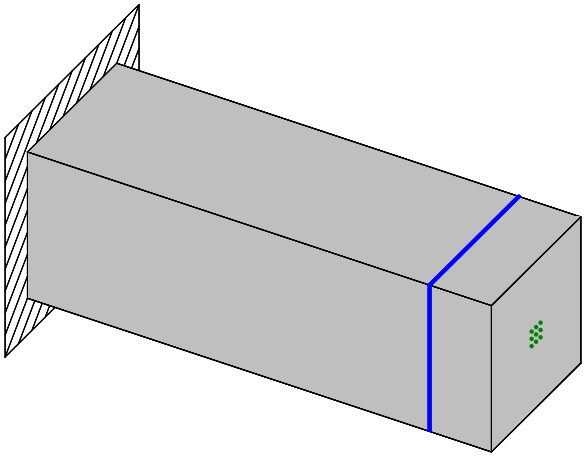
Figure 5.14 – Absolute vertical displacements for design I and II under their worst-case damage.

5.3.3 Fail-safe minimum compliance 3D lattice-type structures

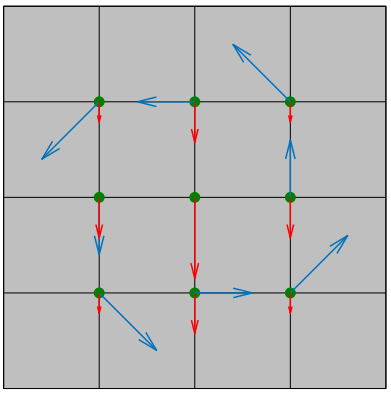
For the shown 3D examples, a design domain of $60 \times 20 \times 20$ elements is used. The filter radius is set to $r = 3$. A lattice-type structure is enforced by using the local volume constraints from section 2.4.3 in conjunction with low local volume fractions k_l in control spheres with a radius of $R = 9$. This procedure breaks shell-like structures down to lattice-type structures which are easier automatically identified and partitioned.

The design domain is again pinned on one side, as shown in figure 5.15a. Loads are distributed to a group of 3×3 central nodes on the opposite side (see figure 5.15b).

For the 3D examples, 300 iterations are performed in stage I and again only 25 iterations in



(a) Design domain with boundary condition, loaded nodes (green) and border of the damage zone (blue)



(b) Detailed view of the loads on the front face for the torsion (blue) and transverse (red) load case

Figure 5.15 – Geometry and loading for the 3D examples. Damage is only applied in the region left of the blue lines in the top figure.

stage II. A global volume constraint in stage II ensures that the shape optimized structure has less or equal volume than the initial structure.

Lattice-type 3D structure and multiple load cases The considered load cases are a torsional load and a downwards pointing transverse tip load with resultant magnitudes of 1 (for the exact distribution, see figure 5.15b). The objective is to minimize the weighted sum of the compliances of the two load cases with weights of 1 for torsion and 0.01 for the transverse load, while a local volume constraint with $k_l = 0.1$ is employed. The resulting structure is shown in figure 5.16a.

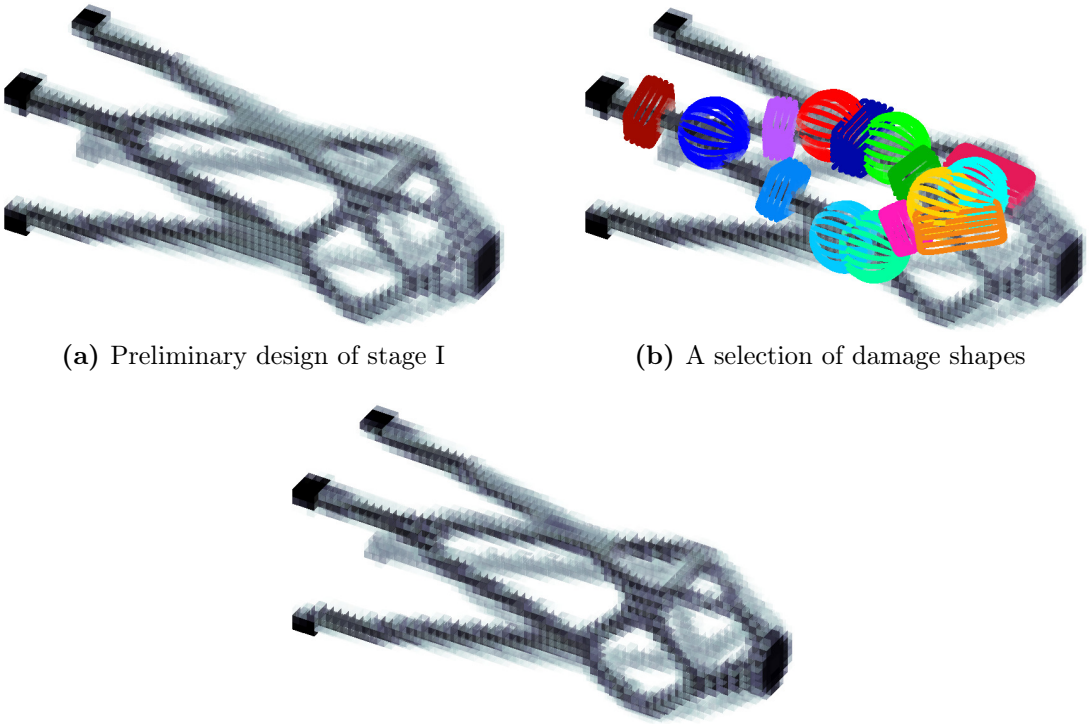
The damage shapes belonging to the identified load paths are shown in figure 5.16b. For a better visibility, only the damage shapes for a quarter of the structure are shown. For the full structure, 51 load paths (30 beams and 21 knots) are identified.

The result from the optimization stage II is shown in figure 5.16c. The change in shape is rather small, but comparing the numerical values in table 5.5 reveals a significant decrease of compliance under the damaged condition to about 62% compared to the initial design. The displacements under torsion for the worst-case damage are shown in figure 5.17.

Lattice-type 3D structure and non-ideal loading This example demonstrates that more drastic changes in shape are possible in the SO of stage II. A structure is at first optimized in stage I for a torsional load only. In optimization stage II not only damage to the load paths is applied but also small transverse forces with a resultant magnitude of 0.1 in up- and downwards directions are additionally applied to simulate a non-ideal load application. The individual load distributions are still the same as shown in figure 5.15b.

A local volume fraction of $k_l = 0.13$ is used. The design from stage I is shown in figure 5.18a. In total 75 load paths (48 beams and 27 knots) are identified. Due to symmetry, only a subset of these shapes is shown in figure 5.18b. In optimization stage II, the beams in the upper and lower region of the part become axially aligned to better absorb the bending introduced by the additional transverse load, as seen in figure 5.18c.

Numerical values for the worst-case and nominal compliances ($\bar{c}$ and c_0, respectively) are



(a) Preliminary design of stage I

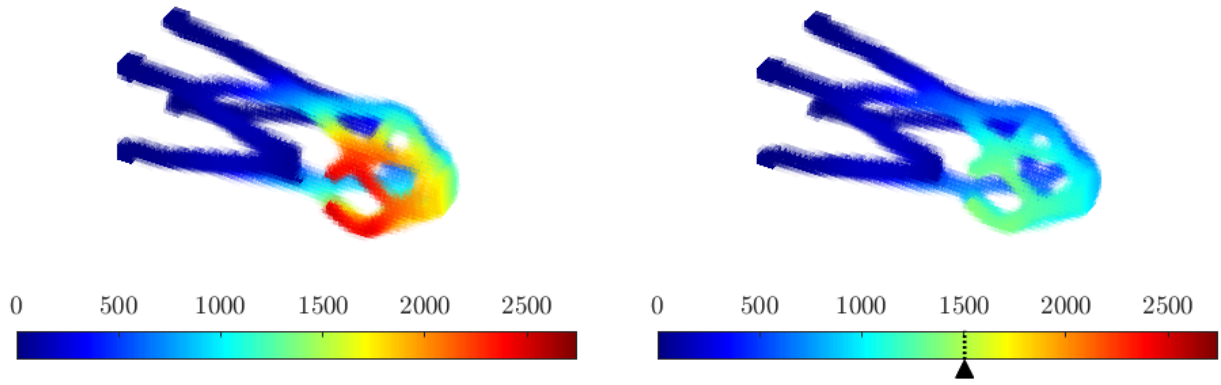
(b) A selection of damage shapes

(c) Design of stage II

Figure 5.16 – Results for compliance minimization with a local volume constraint for a structure under torsional loading and transverse load. For a better visibility, the beams on the far side are shown with less contrast.

Table 5.5 – Numerical results for the 3D example with multiple load cases.

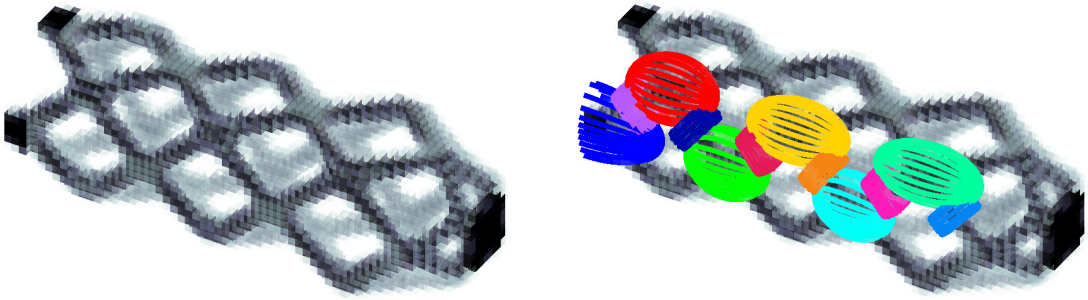
Design	Figure	$\bar{c}$	c_0	ω
Design I	5.16a	23.49	5.19	8.45%
Design II	5.16c	14.64	5.39	8.44%



(a) Design I

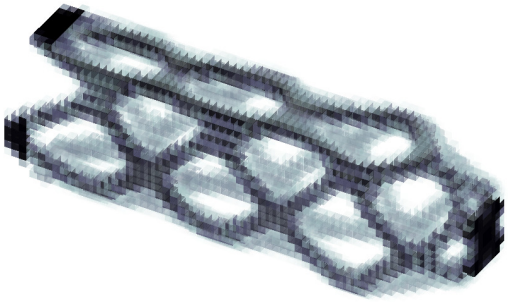
(b) Design II

Figure 5.17 – Absolute displacements of both designs for their worst-case damage under the load case of torsion.



(a) Preliminary design of stage I

(b) A selection of damage shapes



(c) Design of stage II

Figure 5.18 – Results for compliance minimization with a local volume constraint for a structure optimized for torsional loading. In stage II additional transverse forces are acting simultaneously. For a better visibility, the beams on the far side are shown with less contrast.

given in table 5.6. The performance for the non-ideal loading is increased and the overall negative impact of the transverse load is significantly lowered. A more robust design is obtained at the price of a lowered performance for the ideal loading.

Table 5.6 – Numerical results for the 3D example with misaligned loading.

		Ideal load		Non-ideal load		
Design	Figure	$\bar{c}$	c_0	$\bar{c}$	c_0	ω
Design I	5.18a	1.49	0.56	30.66	17.66	11.00%
Design II	5.18c	2.02	1.00	15.56	4.39	11.00%

Since the transverse forces are acting in both directions (up and down), two load cases exist for the non-ideal load. Due to the symmetry of the design, the compliances for both are the same. The values in table 5.6 for the non-ideal loading refer to a single load case, for the mirrored transverse load the same value applies.

5.4 Discussion and further reduction of computational cost

Table 5.7 summarizes the overall expected computational cost for a fail-safe optimization of the example problems in 2D and 3D from section 5.3 using different optimization methods discussed in this work.

The approach from [35], involving the rectangular damage model (RDM), is discussed in section 4.2 and taken as a reference in the following. To be consistent with the results shown in

Table 5.7 – Total number of FE analyses for a fail-safe optimization.

Example	Variant	Figure	Method	Damage cases	Total analyses
2D cantilever	/	n/a	RDM	19 521	3 904 200
	/	4.8b	RDM, GL2	225	45 000
	"robust"	5.5c	proposed SO	24	800
	redundant	5.7c	proposed SO	101	2 725
2D L-bracket	/	n/a	RDM	21 161	4 232 200
	/	4.9b	RDM, GL2	245	49 000
	"robust"	5.10c	proposed SO	16	600
	redundant	5.13c	proposed SO	74	2 050
3D structure	/	n/a	RDM	14 161	4 248 300
	/	n/a	RDM, GL2	2 025	607 500
	multi load case	5.16c	proposed SO	51	1 575
	non-ideal load	5.18c	proposed SO	75	2 175

section 4.2, the same damage zone size of 20×20 elements for the 2D examples is considered. This corresponds to an edge length of one fifth of the shortest design space dimension. Assuming the same relation of sizes, a comparable damage shape size for the shown 3D examples would have a size of $4 \times 4 \times 4$ elements for the RDM. Also, a region with a width of twice the size of a single damage zone is excluded at the tip of the design space from damage considerations.

Thus, using the RDM form [35], a total count of $n_d = 19\,521$ damage scenarios is obtained for the 2D cantilever, $n_d = 21\,161$ for the 2D L-bracket, and $n_d = 14\,161$ for the 3D examples, respectively, which is the number of FE analyses to be run per single optimization iteration. Assuming that the number of iterations is the same as used in this work for a TO without damage considerations (200 iterations for 2D or 300 for 3D), the total number of FE analyses is in the millions (see table 5.7 and section 4.2.1), rendering the method infeasible for industrial application.

With the geometric reduction of damage scenarios of the RDM proposed in [36] and discussed in section 4.2.2, the number of damage cases can be significantly reduced, when using grid-level GL2, which is the recommended choice in the original paper. However, still 225 or 245 analyses have to be done per iteration for the 2D examples and 2025 for the 3D example. The total number of analyses would then be in the order of magnitude of 10^4 to 10^5 for the shown examples (see the entries referring to method "RDM, GL2" in table 5.7). However, these computational savings come at the price that the coarser the damage grid-level is, the more likely it is that a critical damage location is simply disregarded, as revealed by the discussion in section 4.2.2.

For the proposed method for fail-safe optimization, involving a SO stage, the number of failure scenarios to consider is decoupled from the discretization of the FE mesh and only dependent on the number of load-bearing features present. The total number of FE analyses required for the whole optimization is thus composed of a constant number of FE analyses needed in order to obtain the preliminary design (200 in 2D and 300 in 3D) plus a design-dependent number of analyses for each iteration of the second optimization stage, which depends on the number of identified load paths of the structure and is therefore different for every loading scenario considered in this work. Table 5.7 proves, that for the shown examples the total number of analyses required for the proposed two-stage SO method is in the thousands, independently of the design space being modeled in 2D or 3D. Therefore a reduction of the computational cost

of at least three orders of magnitude is achieved when compared to the original RDM.

A huge advantage of the density-based SO method for fail-safe design is, that it converges much faster than a TO "from scratch". Therefore, 25 iterations were sufficient to achieve convergence (see Appendix B.5). But even if the same number of iterations is used for topology and density-based shape optimization, the numerical effort of the sequential approach presented in this work would be lower than with the approach involving coarser damage grid-levels from [36] and much lower than with the original approach involving the full set of damage scenarios of the RDM from [35].

For some compliance based optimization problems, a further reduction of the total computational cost is possible using the proposed load-path-based damage model. Since the knot regions are defined as the regions, where several beams meet, cutting a knot in the considered examples disjoins several beams at once, which has a larger impact on the compliance than a cut of a single beam. Therefore, a worst-case compliance optimization may be conducted with only the knots considered as damage shapes.

The exploitation of this fact is of course only valid for optimization problems, where cutting a knot is equivalent to cutting the corresponding affected beams. A discussion on this aspect is given in Appendix B.2. Also, all beams have to be interrupted by a knot at some point, however, this will mostly be the case, except for the trivial configuration, where the BC regions and loads are connected directly by single straight members.

Consequently, in leaving out the beams, which pose the majority of damage cases in the examples of this work, a further significant reduction of computational cost is achieved with no penalty on the structural performance of the parts. The corresponding reduced sets of damage shapes for the 2D examples treated in section 5.3 are shown in figure 5.19. The obtained final results after the SO stage are visually indistinguishable from the results obtained with the full set of damage shapes and are therefore not repeated[16]. A cross-validation with the full set of damage scenarios has been done with the result, that the worst-case compliances have even improved minimally, probably due to less interference to the gradient of the objective originating from the non-critical damage scenarios concerning beams.

Table 5.8 lists the total number of FE analyses conducted for the redone optimizations of the 2D examples, considering only damage on knot regions in the fail-safe SO stage. In three out of four cases, the computational effort stays even below 1000 analyses.

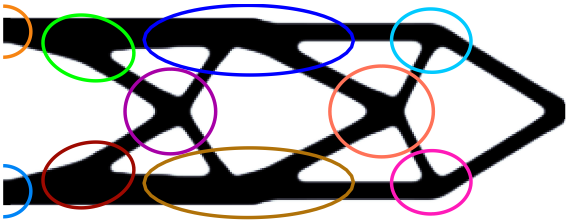
Table 5.8 – Total number of FE analyses for the proposed fail-safe SO considering knots only.

Example	Variant	Damage cases	Total analyses
2D cantilever	"robust"	10	450
	redundant	37	1 125
2D L-bracket	"robust"	5	325
	redundant	26	850

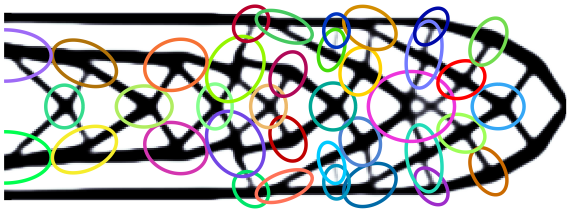
5.5 Results for the alternative formulation

The examples shown in section 5.3 are optimized for a minimum worst-case compliance under a global volume constraint, which corresponds to an optimization for increased damage tolerance.

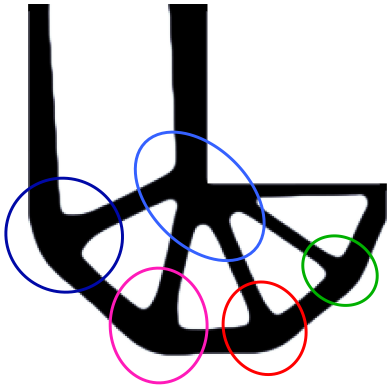
[16]The largest absolute differences in the projected density fields are 0.3 for both cantilever beam designs and 0.1 or 0.8, respectively, for the L-bracket designs.



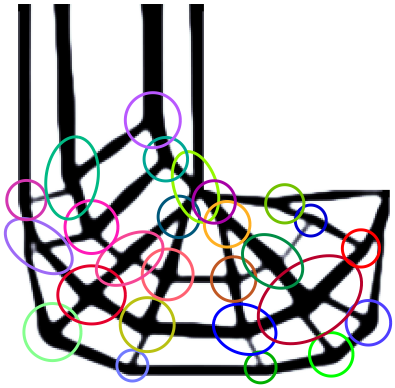
(a) "Robust" cantilever beam (cf. figure 5.5b)



(b) Redundant cantilever beam (cf. figure 5.7b)



(c) "Robust" L-bracket (cf. figure 5.10b)



(d) Redundant L-bracket (cf. figure 5.13b)

Figure 5.19 – Reduced sets of damage shapes for the 2D examples shown in section 5.3.

In this section, also the alternative problem formulation mentioned in section 5.2.5 is studied, where the volume is minimized while a constraint on the maximum allowed worst-case compliance exists, which is closer to the idea of fail-safety.

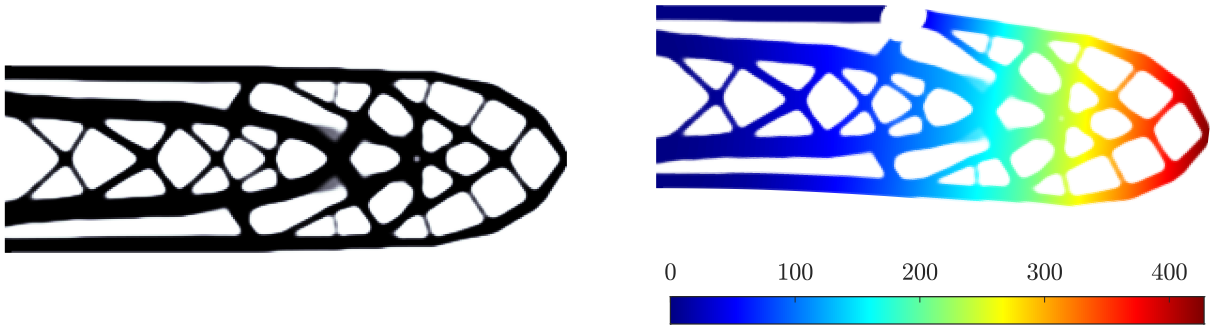
As a challenging optimization task, the lightest part is sought that further cuts down the worst-case compliance of the optimized 2D results from sections 5.3.1 and 5.3.2 to about 75% of the previously obtained values. The optimization task is set up by constraining the aggregated worst-case compliance to the respective threshold value $\bar{c}_t$ and minimizing the volume fraction ω as objective (see equations (5.16)). Since the problem definition involves a conservative KS aggregation for calculating the worst-case compliance, the real obtained worst-case values are expected to be even slightly lower than the allowed threshold values.

Additionally, the method from section 5.4 is used to minimize the computational cost of the fail-safe optimization in considering damage of knot regions only.

The 2D cantilever beam design is shape optimized to a target value for the worst-case compliance of $\bar{c}_t = 450$. Since it is known from the values in section 5.3.1, that the preliminary initial design is far off from this target value, the optimization would start in the infeasible domain, if the shaping DVs $\boldsymbol{\eta}^{II}$ are, as before, initialized with a uniform value of 0.5. A better approach is therefore to start right away with a dilated design in choosing a lower uniform value of e.g. 0.2. Then, the optimization algorithm spends fewer iterations in finding a feasible solution at first.

Fast convergence is observed, such that again only 25 iterations are performed. The optimized cantilever beam is shown in figure 5.20 and has a global volume fraction of $\omega = 0.48$. The nominal, undamaged compliance is $c_0 = 238.45$, while the real worst-case compliance under damage is $\bar{c} = 427.76$.

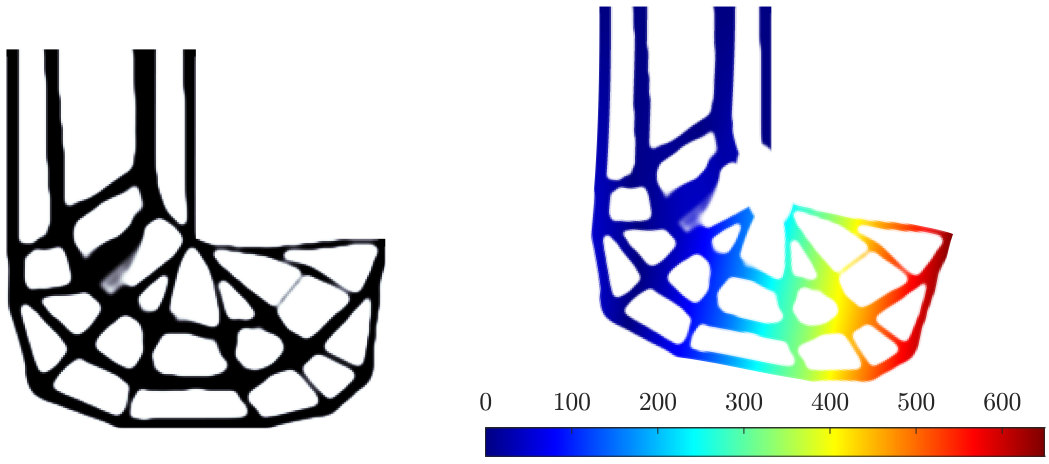
For the 2D L-bracket example, a threshold value of $\bar{c}_t = 650$ is used. The optimization is again initialized with a dilated design with a uniform value of 0.2 for the DVs in $\boldsymbol{\eta}^{II}$. The



(a) Shape optimized design **(b)** Transversal displacement

Figure 5.20 – Shape optimized cantilever beam design and its absolute transversal displacements for the worst-case damage.

optimized design is shown in figure 5.21. The global volume fraction is $\omega = 0.40$, the nominal and worst-case compliances are $c_0 = 149.95$ and $\bar{c} = 648.86$, respectively.



(a) Shape optimized design **(b)** Vertical displacement

Figure 5.21 – Shape optimized L-bracket design and its absolute vertical displacements for the worst-case damage.

Consequently, both structures improved on the worst-case as well as the nominal compliance at the price of an increased amount of solid material (see table 5.9), when compared to the results from section 5.3. The relative increase in volume is 29% for the cantilever beam and 7% for the L-bracket example to achieve the desired drop of the worst-case compliance to less than 75% of the previous optimized level.

However, the optimized designs shown in figures 5.20 and 5.21 have some regions with intermediate projected densities, indicating that the optimization in stage II reached its limit there since for the proposed density-based SO the possible change of the structure is limited.

As a counter-measure, it may be desirable to allow more variation of the preliminary geometry in the SO stage II. This can be easily achieved by recalculating the field $\tilde{\boldsymbol{\varrho}}^I$ from the final DVs $\boldsymbol{\varrho}^I$ of stage I with a larger filter radius than used originally. Then, the zones with a gradual change of the filtered variables in $\tilde{\boldsymbol{\varrho}}^I$ are enlarged and the relevant first projection step in stage II allows a correspondingly larger shift of the outline of the part.

Table 5.9 – Compliance values and volume fractions for the shaped cantilever beam and L-bracket with the alternative formulation.

Example	Figure	$\bar{c}$	c_0	ω
Cantilever beam	5.20a	427.76	238.45	47.96%
L-bracket	5.21a	648.86	149.95	40.42%

5.6 Other applications

The density-based SO presented in this section is also applicable for a use beyond the discussed fail-safe optimizations. Since the SO of stage II has less design freedom than the initial TO in stage I, using the proposed two-stage optimization procedure makes only sense in cases, where the objective function or other modeling aspects are changed in between the two optimization stages.

Two possible use cases for the sequential optimization approach are briefly discussed in the following. However, since both mentioned topics are not in the focus of this work, only a basic outline is given and no further numerical studies are given at this point.

A sequential optimization allows to e.g. handle problems where convergence issues arise if the desired objective is directly tackled by a TO starting from scratch, like it is the case for optimizations involving stress constraints. For such problems, a TO will start in the infeasible domain, since the structure is yet to evolve. As an additional hurdle, violations of the allowed stress may occur even at zero or low density regions [110] such that stress relaxation techniques are required [95].

As an alternative, in [111] a method is presented for obtaining stress constrained designs indirectly by means of a compliance based TO, which offers superior convergence properties. Therein, in order to limit the maximum stresses in a part, length scale control is used to facilitate the evolution of rounded fillets instead of sharp corners, which usually are the source of stress peaks in very localized regions. However, the method of [111] has the drawback, that the obtained results exceed the predefined design space of the model.

Nevertheless, the presented two-stage method is able to follow the same idea in using a compliance optimized starting design from a TO to offer a base design with well-defined geometry for a downstream SO, which purpose is to eliminate stress peaks in sharp corners by rounding them off. Using the proposed method, the part then strictly stays inside the designated design domain, in contrast to the approach from [111].

A second possible use case for the presented two-stage procedure is an optimization towards a pure 0/1 design. With the existing three field TO procedure, many optimization iterations are spent to slowly sharpen the design via a continuation scheme for the projection steepness parameter β (cf. section 2.5.3). In this context, a TO in conjunction with an applied material penalization scheme can be seen as a means to reduce the fraction of elements with intermediate density on a global scale. Due to the filtering step, the outlines of the part will however always stay regions with intermediate densities. To obtain a binary design, a method to drive every single element to a binary state on a local scale is needed afterward. With the proposed method, the SO of stage II can be used for this purpose, if a slight modification is done: The last filtering and projection step needs to be omitted, such that every element can converge to fully solid or void independently of its neighbor. However, it has to be kept in mind, that then also "residual features" in the filtered field (remains from recently vanished structural features) may reappear again. Therefore, the use of the "robust" approach (see section 2.6.1) for the TO in stage I is recommended, since this approach penalizes such residual features.

6 Optimization of Assemblies

In this work, an assembly optimization method involving a density-based TO of components and a simultaneous optimization of joint positions is presented. Large parts of this section are taken from a preceding work of the author, which is published in [59].

In section 6.1, a short introduction to assembly optimization with consideration of joints is given. Therein, two different approaches are compared to augment the previously discussed single-part TO methods to a use in multi-part assembly optimizations. The first discussed approach is a purely density-based approach that motivates the use of a newly developed location-based method, which is presented right afterward and followed throughout the rest of this work.

In the presented location-based method, contrary to existing methods, joints are modeled by patterns of mesh independent springs, allowing arbitrary small movements of the joints during the optimization. Also, as a new feature, the physical size of a joint and of its region of influence is accounted for in the modeling of the force transfer. Each joint additionally comes with a movable non-design space, allowing to impose restrictions on the local geometry concerning e.g. joint mountability. Bolted connections, for example, need holes to be present in the parts to be clamped. In difference to existing methods, the presented method will consider these geometric features with their exact size already at the optimization stage, thus strongly coupling the topology and joint location optimization. Not accounting for that in the simulations means, that holes need to be added in a later machining step, which may severely degrade the load-bearing capacity of the part. Hence, the results of the presented method will consist of exactly the required amount of material and will have exactly the desired amount of discrete joints. The method further goes beyond existing methods for assembly optimization by including fail-safe considerations into the optimization to account for the failure of single or even multiple connections.

Local joint region models for the types of connections in the focus of this work are presented from a geometrical point of view in section 6.2. The mathematical foundation of the overall presented method for assembly optimization is then briefly covered in section 6.3, which lists the DVs and the basic optimization problem setup. Besides that, it is discussed of which parts the global system matrix is composed of and how the individual variables and fields relate to each other.

In-depth details on the joint modeling aspects are given afterward: Section 6.4 explains, how the minimum material and the clearance zones belonging to each joint are implemented as movable non-design spaces by means of a fourth field of modified densities for the TO domains. The modeling of the force transfer in the joints is covered in section 6.5 and includes the insertion of patterns of mesh independent springs and the setup of the corresponding coupling equations.

Objective and constraint functions for a plain assembly optimization without damage considerations are presented in section 6.6, while section 6.7 is dedicated to optimizations considering possible failures of single or multiple joints to obtain assemblies connected in a fail-safe manner. Numerical examples in 2D and 3D for both types of optimizations are given in section 6.8.

6.1 Modeling approaches for assembly optimization problems

A simple approach to tackle the concurrent optimization of multiple parts and their connections relies completely on the plain TO method presented in section 2 and is briefly discussed in section 6.1.1. However, due to some severe downsides arising in high-resolution TO problems,

this approach is not used throughout this work.

Instead, a new approach for assembly optimization is presented in section 6.1.2, which augments the TO framework with its DVs $\boldsymbol{\varrho}$ by joint position variables $\boldsymbol{x}$, thus remedying the deficiencies of the first method. This location-based approach is the recommended method for assembly optimization followed for the remainder of this work.

6.1.1 Assembly optimization using the TO framework only

A method for assembly optimization relying on the TO framework only was presented in [47]. Therein, separate FE meshes exist for every of the n_p parts of the assembly. The FE model is augmented by a regular grid of linear spring elements, each attached between mating nodes of the parts' meshes in an overlap region, as indicated in figure 6.1.

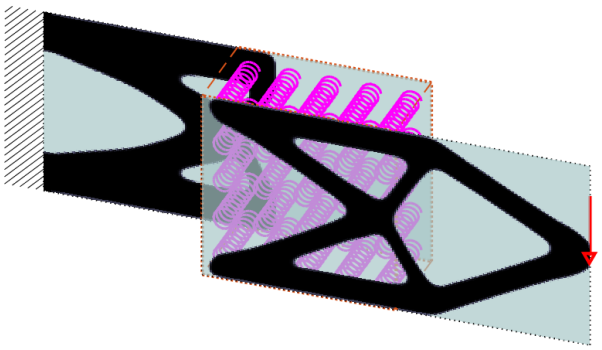


Figure 6.1 – Numerical model for assembly optimization with pure TO methods showing a small exemplary subset of the springs inserted to model joints connecting the two parts.

The springs model joint candidates, allowing a force transfer between the parts of the assembly. Throughout the optimization, similar to the continuum elements, an own "pseudo-density" variable in the range of $[0, 1]$ is assigned to each spring and also the same material penalization scheme (see section 2.3.4) applies to the springs as to the other finite elements in the TO domain. Consequently, this approach requires only minimal adaptation of existing TO codes.

By limiting the total "volume" or count of the springs by an additional volume constraint, which applies solely to these spring DVs, only the most important springs should remain in the final result. These springs are then interpreted as the requisite joints needed to connect the parts of the assembly.

While this method is successfully applied in [47] to a very coarse FE model, in the following it is shown that it does not work as intended for higher resolution models.

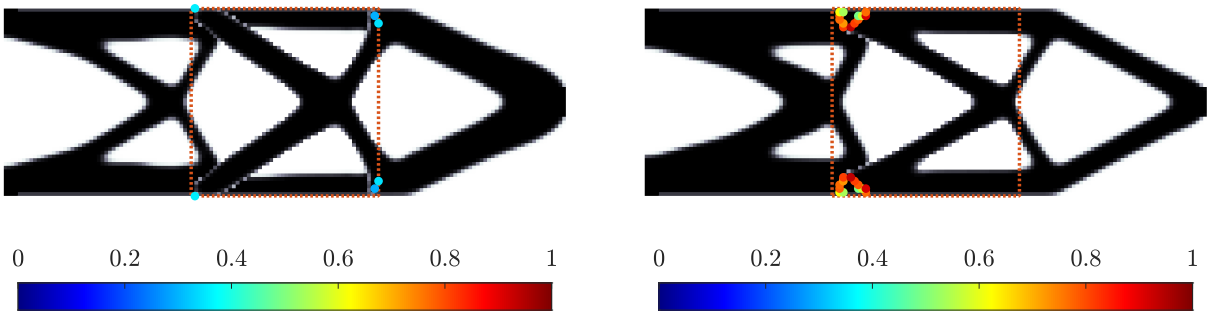
A first major drawback is that the user has no control over the final count of joints present in the result. In having continuous existence variables for each spring and limiting only the total sum (or "volume") of the springs, also partial joints are allowed. Depending on the chosen stiffness of the springs in relation to the stiffness of the parts, despite using a stiffness penalization scheme, these partial joints may still be able to transfer loads sufficiently well. While in the original publication in [47] the springs have converged to a binary state and the exact desired number of distinct joints is obtained, already in a follow-up publication in [112] results with partial joints are presented.

Exemplarily, an optimization of a cantilever beam based on the numerical model shown in figure 6.1 is conducted. A FE discretization comparable to the examples from section 2.7.1

is used, which results in a region of overlap between the two parts with a size of 100×100 elements. Consequently, a grid of 101×101 springs is present to model joint candidates.

At first, the total count of springs is limited to a maximum value of 2 and the result from figure 6.2a is obtained that however does not offer two discrete joints as intended but rather six partial springs (with DV values of approximately a third density for each spring). Furthermore, these springs are located at four distinct regions such that they can rather be interpreted as four separate joints, though only two joints are requested.

Increasing the allowed spring count to a value of e.g. 20, with the intention to obtain a connection involving even more joints, yields the result shown in figure 6.2b. Contrary to the expectation, two clustered groups of springs are obtained that could be interpreted as two large joints.



(a) Limit of 2 springs

(b) Limit of 20 springs

Figure 6.2 – Results for assembly optimizations via TO methods only. The colored dots refer to the positions and DV values of the joint springs in the final result. For a better visibility, springs with low DV values close to zero are not shown.

The results from figure 6.2 thus highlight that with pure TO methods a precise control of the number of joints in an assembly optimization is cumbersome. Also, a simple increase of the allowed spring count is not a viable means to obtain redundant joint patterns if aiming for increased damage tolerance considering single joint failure, ruling this method out for fail-safe assembly optimization.

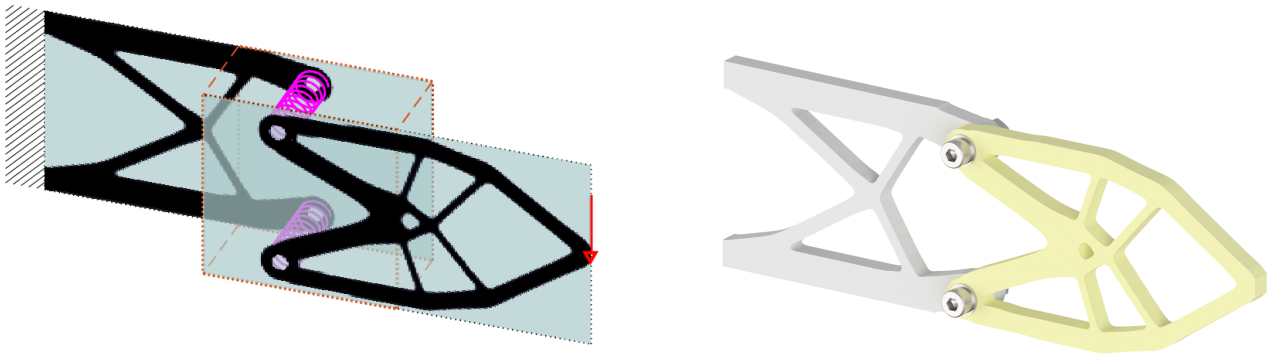
Another major drawback of the discussed assembly optimization method is that the force transfer between the individual parts may involve only single springs forming an isolated node-to-node connection, possibly introducing peak stresses in these regions. Consequently, the size of a joint and its corresponding area of force transfer cannot be modeled. For higher resolution FE meshes, this problem gets worse, as the individual connection springs get smaller in relation to the parts' dimensions and so does their area of influence.

To summarize, optimizing assemblies employing TO methods only yields results that might be taken as inspiration for a manual interpretation of where to place fasteners in an assembly. Nevertheless, this approach lacks detailed modeling of each individual joint and different types of joints cannot be distinguished. Also, the number of joints is not controllable in a straightforward way. These are issues addressed by the newly introduced method for assembly optimization presented in the following.

6.1.2 Assembly optimization involving TO and joint location optimization

The newly presented assembly optimization method differs in the modeling and numerical treatment of joints compared to the previously mentioned method. No pseudo-densities are assigned to the joints. Instead, only the exact number of desired joints is present, each with its own location variables as additional DVs.

As a result, the joints themselves can be placed at arbitrary positions and are not bound to fixed grid positions. To allow control over the spacing between distinct joints, optionally minimum distance constraints may be employed. As an additional feature, the size of the region of force transfer is definable for every joint. Furthermore, it is guaranteed, that every joint is connected to some minimal amount of material. For joints that require mounting holes, like e.g. bolts, these holes are directly considered in the affected mating parts, as indicated in figure 6.3.



(a) Numerical model **(b)** Physical model with fasteners

Figure 6.3 – Example for a simultaneous topology and joint optimization.

The design space of each of the n_p parts is discretized by finite elements. The material distribution for every part is optimized independently but simultaneously, following the TO workflow from section 2. Each part has its own, independent design space, shown as "layers" in figure 6.3a. No nodes are shared between the parts' meshes.

Compared to a standard TO, for the proposed assembly optimization, the vector of DVs is augmented by the locations of the n_j joints' reference points on the contact surface (orange zones in figure 6.3a). Additionally, a solid material zone belongs to every joint (cf. figure 6.4). These material zones are modeled as circular non-design spaces and are included in the material distribution field of each connected part at the location of the joints (see section 6.4).

The fasteners, connecting the parts, are simplified by patterns of springs that model the equivalent stiffness and the size of the force transfer area of the real fastener, which is discussed in detail in section 6.5. Forces between the parts are solely transferred through these spring patterns. The properties and the number of fasteners are predefined and do not change within the optimization. However, the springs move according to the corresponding joint's reference point and have their own independent nodes. Coupling equations link the nodes of each spring to the nodes of the parts, allowing force transfer between the parts.

6.2 Types of joints considered

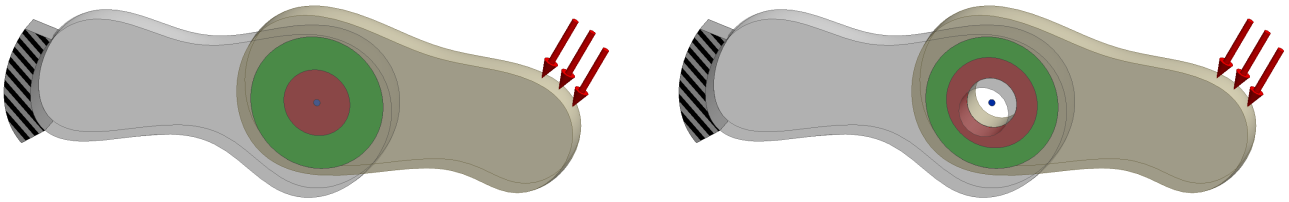
In this work, connections involving axisymmetric geometry are considered. Furthermore, it is distinguished between two different types of such connections (cf. figure 6.4):

- Connections requiring only a minimum amount of material to be present on the mating parts.
- Connections additionally requiring mounting holes to be present in the mating parts.

Spot welds or spot-wise applied adhesives are examples of the first type. Connections involving bolts or rivets are examples of the other type of connection. They require mounting holes of a specific diameter in the connected parts besides some surrounding solid material for the load transfer.

6.2.1 Model of the local joint region

Figure 6.4 shows how the local regions of a joint for the considered connection types are modeled. The red areas designate the "force transfer zones", in which forces act between the two bodies. The green areas are the "minimum material zones" that extend the force transfer zones by an additional amount of solid. Both zones lie on the contact surface of the bodies. The location of the joint on the contact surface is defined by the coordinates of a reference point (blue points in figure 6.4).



(a) A connection without mounting hole **(b)** A connection with mounting hole

Figure 6.4 – Examples for a connection between two bodies. The connection is defined by a reference point (blue), a force transfer zone (red), and an additional material zone (green).

6.3 Extending the TO framework by a location-based joint optimization

In the following, the standard three field TO framework from section 2 is adapted for a use in a multi-part assembly optimization involving the joint model discussed above.

At first, the DVs and the general optimization problem formulation for an assembly optimization are given. Then, the influence of the TO and joint location variables on the assembly of the augmented global system matrix and overall variable processing is outlined. Necessary modifications to the standard three field workflow are described, regarding an adaptation of the filtering step to a multi-part use.

6.3.1 Design variables for assembly optimization

As before, the DVs for the TO are concatenated in a vector $\boldsymbol{\varrho}$ with one entry for every of the n_e elements of the total FE model:

$$\boldsymbol{\varrho} = \begin{bmatrix} \varrho_1 \\ \vdots \\ \varrho_{n_e} \end{bmatrix} \tag{6.1}$$

The DVs of the joints are the locations of their reference points. Without loss of generality, a plane contact surface parallel to the global x-y-plane is considered in this work[17]. The position vector $\boldsymbol{x}^i$ of each of the n_j joints is therefore a vector with two entries (no rotation is considered here):

$$\boldsymbol{x}^i = \begin{bmatrix} x^i \\ y^i \end{bmatrix}, \qquad i = 1, \ldots, n_j \tag{6.2}$$

The combined design vector $\boldsymbol{x}$ contains all of the joints' positions:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}^1 \\ \vdots \\ \boldsymbol{x}^{n_j} \end{bmatrix} \tag{6.3}$$

6.3.2 Optimization problem formulation

The optimization problem for a simultaneous topology and joint location optimization is obtained by augmenting the plain TO optimization problem from equations (2.1) to:

$$\min_{\boldsymbol{\varrho},\boldsymbol{x}} \Phi\left(\tilde{\boldsymbol{u}}(\boldsymbol{\varrho},\boldsymbol{x})\right) \tag{6.4a}$$

$$\text{s.t.} \quad h_i(\boldsymbol{\varrho},\boldsymbol{x}) \leq 0 \qquad \forall i \tag{6.4b}$$

$$0 \leq \varrho_j \leq 1 \qquad j = 1, \ldots, n_e \tag{6.4c}$$

$$\left.\begin{aligned} x_l \leq x^k \leq x_u \\ y_l \leq y^k \leq y_u \end{aligned}\right\} \qquad k = 1, \ldots, n_j \tag{6.4d}$$

$$\tilde{\boldsymbol{K}}(\boldsymbol{\varrho},\boldsymbol{x})\tilde{\boldsymbol{u}} = \tilde{\boldsymbol{f}} \tag{6.4e}$$

In contrast to the plain TO problem from section 2.1, the objective Φ, the constraint functions h_i, and the system matrix $\tilde{\boldsymbol{K}}$ now additionally depend on the joint position DVs in $\boldsymbol{x}$.

The objective function Φ is considered to be dependent on an augmented displacement vector $\tilde{\boldsymbol{u}}$ that, in turn, is the solution to the equilibrium equation (6.4e) involving an augmented system matrix $\tilde{\boldsymbol{K}}$, whose composition is discussed in section 6.3.3.

The optimization is further constrained by one or more constraint functions h_i. How to adapt the objective and constraint functions from section 2.4 to an assembly optimization problem is discussed in section 6.6. There, also two additional constraints for controlling the properties of the joints are presented.

Equations (6.4c) and (6.4d) enforce bounds on the values of the DVs. The material DVs $\boldsymbol{\varrho}$ are, as before, limited to a value between 0 and 1. The position variables in $\boldsymbol{x}$ are constrained by upper and lower bounds on their coordinates, such that the joints, including their material zones, stay inside the designated contact surface area of the parts.

6.3.3 Augmented system matrix

The augmented stiffness matrix $\tilde{\boldsymbol{K}}$ from equation (6.4e) is composed of the stiffness contribution $\boldsymbol{K}_m$, originating from the material of the parts, the matrix $\boldsymbol{K}_c$, modeling the connection stiffness

[17] For other orientations or coordinate system types, a coordinate transformation can be performed.

of the joints, and linear coupling equations, grouped in a matrix $\boldsymbol{G}$:

$$\tilde{\boldsymbol{K}}\tilde{\boldsymbol{u}} = \tilde{\boldsymbol{f}} \tag{6.5a}$$

$$\left(\boldsymbol{K}_m(\boldsymbol{\varrho}, \boldsymbol{x}) + \boldsymbol{K}_c + \boldsymbol{G}(\boldsymbol{x})\right)\tilde{\boldsymbol{u}} = \tilde{\boldsymbol{f}} \tag{6.5b}$$

Since every joint comes with a material zone as non-design space, the material part $\boldsymbol{K}_m$ is not only dependent on the material DVs $\boldsymbol{\varrho}$ but also on the joint locations, stored in $\boldsymbol{x}$. The matrix $\boldsymbol{K}_c$ of the joints is constant since the stiffness properties of each connection are predefined. The coupling equations in $\boldsymbol{G}$ control where the individual joints are placed on the parts and thus define where a force transfer is possible. Therefore, $\boldsymbol{G}$ is solely dependent on the location variables $\boldsymbol{x}$.

All parts and all joints are independent of each other and do not share degrees of freedom. The individual stiffness matrices $\boldsymbol{K}_m^i$ of each part i are therefore placed on the diagonal of $\tilde{\boldsymbol{K}}$. The same holds for the stiffness matrices $\boldsymbol{K}_c^j$ of each joint j. The augmented stiffness matrix $\tilde{\boldsymbol{K}}$ has therefore the shape shown in equation (6.6). Only the linear coupling equations in $\boldsymbol{G}$ link the degrees of freedom of the joints and the parts, allowing a force transfer between the parts.

$$\tilde{\boldsymbol{K}} = \begin{bmatrix} \begin{bmatrix} \boldsymbol{K}_m^1 & & \boldsymbol{0} \\ & \ddots & \\ \boldsymbol{0} & & \boldsymbol{K}_m^{n_p} \end{bmatrix} & \boldsymbol{0} & \begin{bmatrix} \\ \\ \boldsymbol{G}^T \\ \\ \\ \end{bmatrix} \\ \boldsymbol{0} & \begin{bmatrix} \boldsymbol{K}_c^1 & & \boldsymbol{0} \\ & \ddots & \\ \boldsymbol{0} & & \boldsymbol{K}_c^{n_j} \end{bmatrix} & \\ \begin{bmatrix} & \boldsymbol{G} & \end{bmatrix} & & \boldsymbol{0} \end{bmatrix} \tag{6.6}$$

The augmented displacement vector $\tilde{\boldsymbol{u}}$ contains the nodal displacements of the parts $\boldsymbol{u}_m$ and the joints $\boldsymbol{u}_c$, as well as the Lagrange-multipliers $\boldsymbol{\lambda}_c$ for the coupling equations:

$$\tilde{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{u}_m \\ \boldsymbol{u}_c \\ \boldsymbol{\lambda}_c \end{bmatrix}, \text{with:} \qquad \boldsymbol{u}_m = \begin{bmatrix} \boldsymbol{u}_m^1 \\ \vdots \\ \boldsymbol{u}_m^{n_p} \end{bmatrix}, \quad \boldsymbol{u}_c = \begin{bmatrix} \boldsymbol{u}_c^1 \\ \vdots \\ \boldsymbol{u}_c^{n_j} \end{bmatrix}, \quad \boldsymbol{\lambda}_c = \begin{bmatrix} \boldsymbol{\lambda}_c^1 \\ \vdots \\ \boldsymbol{\lambda}_c^{n_j} \end{bmatrix} \tag{6.7}$$

There are as many Lagrange multipliers in $\boldsymbol{\lambda}_c$ as entries in $\boldsymbol{u}_c$, since all degrees of freedom of the joints are coupled by an own coupling equation.

Assuming, that external loads only act on the nodes of the parts, the augmented force vector $\tilde{\boldsymbol{f}}$ is:

$$\tilde{\boldsymbol{f}} = \begin{bmatrix} \boldsymbol{f}_m \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \text{with:} \qquad \boldsymbol{f}_m = \begin{bmatrix} \boldsymbol{f}_m^1 \\ \vdots \\ \boldsymbol{f}_m^{n_p} \end{bmatrix} \tag{6.8}$$

6.3.4 Data processing for system matrix composition

Figure 6.5 gives an extended overview of how the different variables inside an assembly optimization are related to each other and how they affect the different parts of the augmented system matrix $\tilde{\boldsymbol{K}}$.

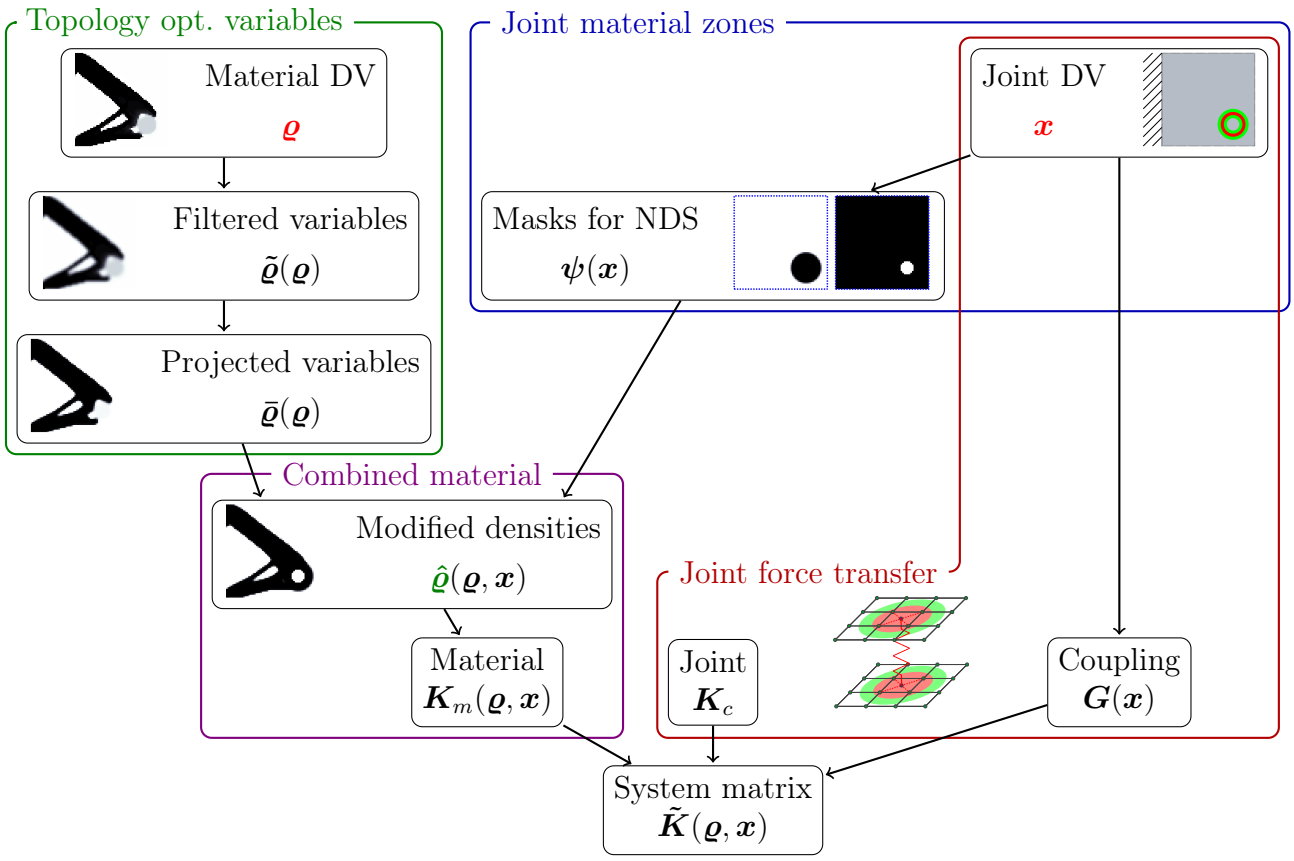


Figure 6.5 – Extended data processing diagram for assembly optimization.

On the TO side, the material DVs $\boldsymbol{\varrho}$ yield filtered and projected variables, which is covered in section 6.3.5. The joint position variables $\boldsymbol{x}$ are needed to define mask vectors $\boldsymbol{\psi}$ that are used to employ the material zones of each joint as a non-design space, as explained in the following section 6.4. A field of modified densities $\hat{\boldsymbol{\varrho}}$ combines the material distribution obtained by TO with the NDS of the joints and is the input for the material part $\boldsymbol{K}_m$ of the global system matrix.

On the joint optimization side, section 6.5 shows how the joints' position variables $\boldsymbol{x}$ influence the coupling matrix $\boldsymbol{G}$ that links the connection springs to the parts. The joints themselves are modeled by patterns of springs with a constant predefined stiffness. All joints are combined to the connection part $\boldsymbol{K}_c$ of the global stiffness matrix.

6.3.5 The basic three TO fields in an assembly optimization

As shown in figure 6.5, the variable processing workflow on the TO side in the proposed assembly optimization method follows the standard three field TO approach from section 2.3. However, the filtering step has to be slightly adapted.

For a multi-component TO, the filtering has to be applied on a per-part basis, otherwise, material would be transferred between different parts in areas, where the FE meshes overlap or contact each other, hindering the parts to evolve independently. Therefore, the filter equation (2.2) from section 2.3.1 is altered, such that all n_p parts are filtered separately.

The meshes of every part are treated as independent "layers" and the filter equation is only evaluated over a set $\mathbb{L}_k$ containing the element indices of the current part k. For an assembly

optimization, the filter equation (2.2a) is therefore replaced by:

$$\tilde{\varrho}_i = \frac{\sum_{j \in \mathbb{L}_k} w(\boldsymbol{c}^j - \boldsymbol{c}^i) v_j \varrho_j}{\sum_{j \in \mathbb{L}_k} w(\boldsymbol{c}^j - \boldsymbol{c}^i) v_j}, \qquad k = 1, \ldots, n_p \tag{6.9}$$

The projection step of the three field TO described in section 2.3.2 however stays unmodified.

6.3.6 Modified densities in the context of assembly optimization

In the proposed method, also a fourth field of modified densities $\hat{\boldsymbol{\varrho}}$ is needed, as shown in figure 6.5. Its purpose is to employ the moving non-design spaces (NDS), originating from the material and clearance zones of the joints, into the material distribution field obtained by TO (for details, see section 6.4), thus creating a link between the material and the position DVs.

The modified densities $\hat{\boldsymbol{\varrho}}$ are therefore dependent on the projected TO variables $\bar{\boldsymbol{\varrho}}$ and on the joints' locations in $\boldsymbol{x}$. The NDS are defined by (at least) one combined mask vector $\boldsymbol{\psi}$ calculated based on the joints' locations:

$$\hat{\boldsymbol{\varrho}} = \hat{\boldsymbol{\varrho}}\left(\bar{\boldsymbol{\varrho}}, \boldsymbol{\psi}(\boldsymbol{x})\right) \tag{6.10}$$

This approach corresponds to a generalization of the modified densities discussed in section 4, where mask vectors are considered as constant and only used to remove material. For the use in assembly optimization, masks are also used to bring in material.

The different types of moving NDS handled by the modified density field are discussed in detail in section 6.4. All joints' NDS are projected at once into $\hat{\boldsymbol{\varrho}}$, such that for a standard assembly optimization task only one field of modified densities is present, as shown in the simplified variable processing diagram in figure 6.6.

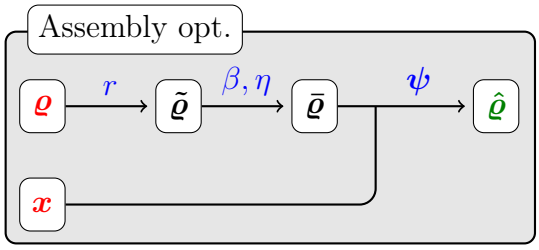


Figure 6.6 – Relation of variables for an assembly optimization including topology and joint location optimization.

The modified densities $\hat{\boldsymbol{\varrho}}$ are the pseudo-densities considered for the stiffness evaluation of every finite element in the analysis, according to the chosen material penalization model (cf. section 2.3.4).

6.4 Modified densities for moving non-design spaces

The non-design spaces (NDS) are used to employ solid material zones or holes of specific sizes for every joint. Only circular (or cylindrical) features are considered in this work (cf. figure 6.4). The NDS follow the joints' reference points and therefore need to be continuously movable, independently of the discretization of the underlying FE mesh.

The NDS are thus defined by parametric shapes that are mapped to a mask vector $\boldsymbol{\psi}$, following the approach presented in section 4.3.2. The resulting combined mask $\boldsymbol{\psi}$ is composed of individual masks $\boldsymbol{\psi}^i$, obtained for every joint i. Section 6.4.1 describes, how a single mask is calculated, section 6.4.2 covers the combination of masks.

The masks can then further be used to either remove material (denoted by ${}^{-}\boldsymbol{\psi}$), add material (denoted by ${}^{+}\boldsymbol{\psi}$), or do both by subsequent application. These three cases are discussed in sections 6.4.3 – 6.4.5.

6.4.1 Mask for a single axisymmetric feature

The mask vector $\boldsymbol{\psi}^i$ for joint i has one entry for every of the n_e finite elements of the model and has therefore the same size as the material design vector $\boldsymbol{\varrho}$.

The joint's mask will define a circular shape (or cylindrical shape in 3D) around the joint's reference point $\boldsymbol{x}^i$ in the x-y-plane, which is considered as the contact plane here. The entries of $\boldsymbol{\psi}^i$ are calculated using the already given equations (4.11) – (4.14). For a circular feature aligned with the axes of the global coordinate system, $\boldsymbol{V}$ is the identity matrix and the semi-axes a and b equal the desired NDS radius r'. Thus, the equations for the mask values ψ^i_j and the shape outline function E (with the shape parameter $q = 2$) simplify to:

$$\psi^i_j = \frac{\tanh\left(\alpha E(\boldsymbol{c}^j - \boldsymbol{x}^i)\right) + 1}{2}, \qquad E(\boldsymbol{x}') = \left(\frac{x'}{r'}\right)^2 + \left(\frac{y'}{r'}\right)^2 - 1 \tag{6.11}$$

Depending on for which geometric feature the mask is used, r' is either the radius of the solid material zone or the radius of a hole. The parameter α controls the sharpness of the projection of the circular shape on the density field, a value of $\alpha = 10$ is used in this work (cf. section 4.3.2).

The equations in (6.11) are differentiable with respect to the joint's reference point coordinates $\boldsymbol{x}^i$. The partial derivatives $\frac{\partial \boldsymbol{\psi}^i}{\partial x^i}$ and $\frac{\partial \boldsymbol{\psi}^i}{\partial y^i}$ are given in Appendix E.1.

6.4.2 Combined masks and complementary masks

The masks $\boldsymbol{\psi}^i$, obtained for every joint i, are composed into a resulting mask $\boldsymbol{\psi}$ by element-wise multiplication:

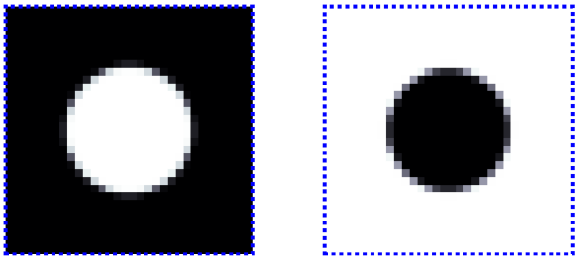
$$\boldsymbol{\psi} = \prod_{i=1}^{n_j} \boldsymbol{\psi}^i \tag{6.12}$$

The derivatives of the combined mask with respect to the coordinates of the reference point of the i-th joint are given in Appendix E.2.

The complementary mask is obtained by:

$$\boldsymbol{\psi}_{comp} = \boldsymbol{I} - \boldsymbol{\psi} \tag{6.13}$$

Figure 6.7 shows a mask and its complementary mask for a single circle with a radius of 8 inside a 32×32 domain. The parameter α is set to a value of 10.



(a) Mask $\boldsymbol{\psi}$ **(b)** Complementary mask $\boldsymbol{\psi}_{comp}$

Figure 6.7 – A mask and its complementary mask.

6.4.3 Modified densities for void NDS

In practical applications, holes have to be present to mount fasteners like bolts or rivets. Following the method presented in section 4.3.1, a multiplicative mask is used to model the holes in the structure. The holes move according to the reference points and always enforce that the affected regions are kept free of solid material.

With the combined mask vector defined in equation (6.12), several holes can be introduced at once into the modified density field $\hat{\boldsymbol{\varrho}}$ by element-wise multiplication (cf. equation (4.10a)):

$$\hat{\boldsymbol{\varrho}} = \bar{\boldsymbol{\varrho}} \circ {}^{-}\boldsymbol{\psi} \tag{6.14}$$

The values of the mask ${}^{-}\boldsymbol{\psi}$, designated to remove material, linearly interpolate the modified density values between zero and the value of the original projected variables in $\bar{\boldsymbol{\varrho}}$.

The joint's mask vectors are obtained by applying the equations in (6.11) with the desired hole radius. The sensitivities with respect to the projected densities and the joints' locations are given in Appendix E.3.

The application of the mask from figure 6.7a to enforce a hole according to equation (6.14) in an exemplary gradual density field is shown on the left side of figure 6.8.

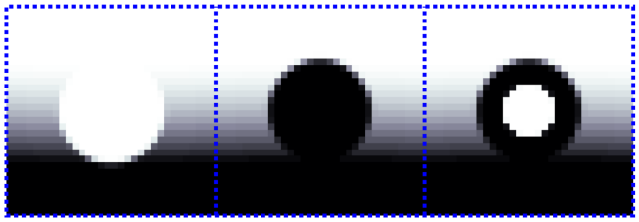


Figure 6.8 – Application of masks to produce a hole, a circle, and a ring in a sample density field.

6.4.4 Modified densities for solid NDS

Let $\bar{\boldsymbol{\varrho}}_{comp}$ and $\hat{\boldsymbol{\varrho}}_{comp}$ denote the complementary field of the projected variables and modified densities, where solid and void regions are interchanged:

$$\bar{\boldsymbol{\varrho}}_{comp} = \boldsymbol{I} - \bar{\boldsymbol{\varrho}}, \qquad \hat{\boldsymbol{\varrho}}_{comp} = \boldsymbol{I} - \hat{\boldsymbol{\varrho}} \tag{6.15}$$

The opposite effect of enforcing voids is achieved if the method from section 6.4.3 is applied to the complementary projected variable field:

$$\hat{\boldsymbol{\varrho}}_{comp} = \bar{\boldsymbol{\varrho}}_{comp} \circ {}^{+}\boldsymbol{\psi} \tag{6.16}$$

If the result is then inverted again, a material zone is added to the density field. Therefore, the mask is now referred to as ${}^{+}\boldsymbol{\psi}$, since its purpose is to bring in material. Rearranging equation (6.16) by using (6.15) yields:

$$\begin{aligned}
\hat{\boldsymbol{\varrho}} &= \boldsymbol{I} - \bar{\boldsymbol{\varrho}}_{comp} \circ {}^{+}\boldsymbol{\psi} && \text{(6.17a)}\\
&= \boldsymbol{I} - (\boldsymbol{I} - \bar{\boldsymbol{\varrho}}) \circ {}^{+}\boldsymbol{\psi} && \text{(6.17b)}\\
&= (\boldsymbol{I} - {}^{+}\boldsymbol{\psi}) + \bar{\boldsymbol{\varrho}} \circ {}^{+}\boldsymbol{\psi} && \text{(6.17c)}\\
&= {}^{+}\boldsymbol{\psi}_{comp} + \bar{\boldsymbol{\varrho}} \circ {}^{+}\boldsymbol{\psi} && \text{(6.17d)}
\end{aligned}$$

Equation (6.17c) shows, that the components of the mask ${}^{+}\boldsymbol{\psi}$ linearly interpolate the modified densities $\hat{\boldsymbol{\varrho}}$ between the value one and the original values in $\bar{\boldsymbol{\varrho}}$. The corresponding derivatives are given in Appendix E.4.

The joints' mask values are again obtained by the equations in (6.11), but in this case, the desired solid material zone radius is used as parameter r'.

The application of the masks from figure 6.7, according to equation (6.17d), adds a material circle to the gradual sample field as shown in the middle of figure 6.8.

6.4.5 Modified densities for ring-shaped solid NDS

Subsequent application of the procedures from sections 6.4.4 and 6.4.3 yields material rings (or hollow cylinders in 3D): First a circular material zone with a larger outer radius is added and then the material inside an inner radius zone is removed:

$$\hat{\boldsymbol{\varrho}} = ({}^{+}\boldsymbol{\psi}_{comp} + \bar{\boldsymbol{\varrho}} \circ {}^{+}\boldsymbol{\psi}) \circ {}^{-}\boldsymbol{\psi} \tag{6.18}$$

The derivatives of the above equation are given in Appendix E.5. Using the masks from figure 6.7 to add material and a mask with half the radius to remove material from the sample density field yields the result shown on the right side of figure 6.8.

6.5 Force transfer in connections

The force transfer between the connected parts is enabled by inserting stiff springs, located inside the force transfer zone of every joint, as demonstrated in figure 6.9. Just like the NDS discussed in section 6.4, these springs also follow the joints' reference points and need to be continuously movable across the FE mesh.

6.5.1 Mesh independent springs

For every spring, a new dependent "slave" node is inserted at the exact location of the spring on the contact surface of every part (see figure 6.9). The dependent nodes are coupled to the nodes of the corresponding element, which act as "master" nodes, to ensure displacement compatibility.

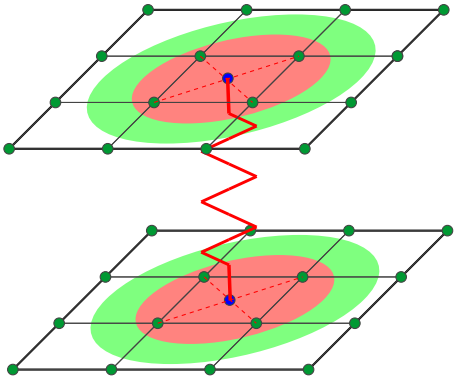


Figure 6.9 – Mesh independent spring (in all DOF) with dependent nodes (blue) coupled to the nodes of the parts' meshes.

Using this method, the meshes of the connected parts do not need to be congruent at the contact surface.

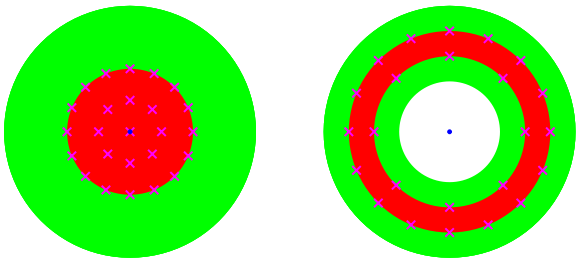
The inserted springs have the same stiffness in all degrees of freedom to model isotropic force transfer. The implementation of orthotropic behavior with e.g. different normal than tangential stiffness is straightforward.

6.5.2 Spring patterns

Since using only one spring per joint lacks the modeling of the physical size of the area where a joint can transfer load, distributed patterns of springs are used for every connection. The whole pattern will move as a unit, according to the movement of the joint's reference point.

For the examples considered in this work, concentric patterns of springs are used. Figure 6.10a shows the model used for connections that do not require a hole, like e.g. spot welds. As NDS, a solid material zone (green) is present for this type of connection. A subarea of the material zone is the force transfer zone (red), over which 25 springs (including one at the central reference point) are distributed. For connections requiring a hole, the pattern from figure 6.10b, consisting of 24 springs (no spring at the center), is used.

For both cases, the resultant stiffness of the spring patterns has the same value k_c. The stiffness k_c is chosen such, that it models the stiffness of the real connection element. In this work, it is assumed, that the total compliance of the assembly is dominated by the compliance of the individual parts and that the connectors are comparably stiff. Therefore, stiff springs are used.



(a) Circular pattern **(b)** Ring pattern

Figure 6.10 – NDS zone (green), force transfer zone (red), reference points (blue), and spring patterns (magenta) used in the examples.

6.5.3 Coupling equations

The coupling of the dependent spring nodes to the master nodes of the parts (see figure 6.9) is described via linear equations, linking the displacements of the nodes [113].

Compatibility of the displacements is ensured if the dependent node of the joint's spring is linked to the corresponding element's nodal displacements via the isoparametric weight coefficients:

$$\boldsymbol{g}^i = \sum_{j=1}^{n_n} w_j \boldsymbol{u}_m^j - \boldsymbol{u}_c^i = \boldsymbol{0} \tag{6.19}$$

Here, $\boldsymbol{u}_c^i$ is a slave node's displacement vector belonging to a connector spring of the i-th joint, and $\boldsymbol{u}_m^j$ are the corresponding element's j-th nodal displacement vectors.

The weights w_j in equation (6.19) are the values of the n_n element's shape functions at the position of the slave node. The values are obtained by transforming the global coordinates of the slave node to the local coordinates $\boldsymbol{\xi}$ of the isoparametric element, which are then a function of the joint's reference point $\boldsymbol{x}^i$. The shape functions N_j (see Appendix C.5) are evaluated at these local coordinates:

$$w_j = N_j(\boldsymbol{\xi}(\boldsymbol{x}^i)), \qquad j \in \{1, \ldots, n_n\} \tag{6.20}$$

Equation (6.19) yields two (or three in 3D) linear equations per spring of the pattern of each joint for all parts affected by the connection. The coefficients of all these coupling equations are put as row vectors into a couping matrix $\boldsymbol{G}$:

$$\begin{bmatrix} \boldsymbol{g}^1(\boldsymbol{x}^1) \\ \vdots \\ \boldsymbol{g}^{n_j}(\boldsymbol{x}^{n_j}) \end{bmatrix} = \boldsymbol{G}(\boldsymbol{x}) \begin{bmatrix} \boldsymbol{u}_m \\ \boldsymbol{u}_c \end{bmatrix} = \boldsymbol{0} \tag{6.21}$$

The coefficient matrix $\boldsymbol{G}$ links all dependent slave connector displacements $\boldsymbol{u}_c$ to the master material displacements $\boldsymbol{u}_m$ of the parts.

The derivative of $\boldsymbol{G}$ uses the derivatives of the isoparametric shape functions:

$$\frac{dw_j}{dx_i} = \sum_k \frac{dN_j}{d\xi_k} \frac{d\xi_k}{dx_i} \tag{6.22}$$

Lagrange multipliers $\boldsymbol{\lambda}_c$ are used to incorporate the coupling equations into the system matrix[18] (see section 6.3.3). The Lagrange multipliers are then the constraint forces needed to keep the slave nodes at the desired positions (cf. section 6.6.4).

[18] Using Lagrange multipliers increases the size of the system matrix by introducing additional artificial DOFs to the system. Thus, also the associated computational cost for solving the system is increased. A numerically more efficient way to deal with linear constraint equations is to apply direct elimination techniques, which even decrease the size of the system matrix and are recommended for a numerical implementation. However, for presenting the theory of the proposed assembly optimization method, not using direct elimination is preferred, due to the clearly separated structure of the augmented system matrix from equation (6.6). Also, in using Lagrange multipliers, the setup of the adjoint system for the joint force constraint in section 6.6.4 is simplified.

The matrix $\boldsymbol{G}$ contains isoparametric weights in the range of $[0, 1]$ as coefficients for the master DOFs and weights of -1 for the slave connector DOFs (cf. equation (6.19)). In order to not jeopardize the condition of the augmented global system matrix $\tilde{\boldsymbol{K}}$ shown in equation (6.6), a scaling of the coefficients in $\boldsymbol{G}$ should be considered if the order of magnitude of the entries of the other matrices $\boldsymbol{K}_m$ and $\boldsymbol{K}_c$ is much larger than the one for the entries of $\boldsymbol{G}$. If coefficient scaling is applied to $\boldsymbol{G}$, the homogeneous constraint equations from (6.19) will still be satisfied, however, the values in $\boldsymbol{\lambda}_c$ will then be the inversely scaled versions of the constraint forces.

6.6 Objectives and constraints in a plain assembly optimization

The objective and constraint functions known from plain TO presented in section 2.4 are in the following adapted for a use in assembly optimizations.

Also, two additional constraints referring to specific properties of joints are introduced, namely a minimum distance and a maximum force constraint.

6.6.1 Compliance objective

The definition of the compliance from equation (2.8) is applied to the augmented force and displacement vectors:

$$\Phi = c\left(\boldsymbol{\varrho}, \boldsymbol{x}\right) = \tilde{\boldsymbol{f}}^T \tilde{\boldsymbol{u}} \tag{6.23}$$

Exploiting the structure of the augmented stiffness matrix from (6.6) and expanding equation (6.5b), the compliance within an assembly optimization can be expressed as:

$$c = \boldsymbol{u}_m^T \boldsymbol{K}_m \boldsymbol{u}_m + \boldsymbol{u}_c^T \boldsymbol{K}_c \boldsymbol{u}_c + \begin{bmatrix} \boldsymbol{u}_m^T & \boldsymbol{u}_c^T \end{bmatrix} \boldsymbol{G}^T \boldsymbol{\lambda}_c + \boldsymbol{\lambda}_c^T \boldsymbol{G} \begin{bmatrix} \boldsymbol{u}_m \\ \boldsymbol{u}_c \end{bmatrix} \tag{6.24}$$

The linear coupling equations linking the displacements of the material nodes with the joint nodes are homogeneous, i.e. they sum up to zero:

$$\boldsymbol{G} \begin{bmatrix} \boldsymbol{u}_m \\ \boldsymbol{u}_c \end{bmatrix} = \boldsymbol{0} \tag{6.25}$$

Therefore, the last two summands of equation (6.24) do not contribute to the compliance (which means that the coupling equations do not store elastic energy). Hence, the compliance can be subdivided into the compliance contributions of every part plus the contributions of every joint:

$$\begin{aligned} c &= \boldsymbol{u}_m^T \boldsymbol{K}_m \boldsymbol{u}_m + \boldsymbol{u}_c^T \boldsymbol{K}_c \boldsymbol{u}_c && \text{(6.26a)} \\ &= \sum_{i=1}^{n_p} \boldsymbol{u}_m^{i\,T} \boldsymbol{K}_m^i \boldsymbol{u}_m^i + \sum_{i=1}^{n_j} \boldsymbol{u}_c^{i\,T} \boldsymbol{K}_c^i \boldsymbol{u}_c^i && \text{(6.26b)} \\ &= \sum_{i=1}^{n_p} c_m^i + \sum_{i=1}^{n_j} c_c^i && \text{(6.26c)} \\ &= c_m + c_c && \text{(6.26d)} \end{aligned}$$

The total derivative of the compliance with respect to the material design variables consists of four partial derivatives[19] (cf. the four density fields in figure 6.6):

$$\frac{dc}{d\varrho_i} = \sum_j \frac{\partial c}{\partial \hat{\varrho}_j} \frac{\partial \hat{\varrho}_j}{\partial \bar{\varrho}_j} \frac{\partial \bar{\varrho}_j}{\partial \tilde{\varrho}_j} \frac{\partial \tilde{\varrho}_j}{\partial \varrho_i} \tag{6.27}$$

The first partial derivative is again obtained by adjoint sensitivity analysis (see Appendix D.2):

$$\frac{\partial c}{\partial \hat{\varrho}_j} = -\tilde{\boldsymbol{u}}^T \frac{\partial \tilde{\boldsymbol{K}}}{\partial \hat{\varrho}_j} \tilde{\boldsymbol{u}} = -\tilde{\boldsymbol{u}}^T \frac{\partial \boldsymbol{K}_m}{\partial \hat{\varrho}_j} \tilde{\boldsymbol{u}} = -\boldsymbol{u}_m^T \frac{\partial \boldsymbol{K}_m}{\partial \hat{\varrho}_j} \boldsymbol{u}_m \tag{6.28}$$

Equation (6.28) utilizes that only the material part of $\tilde{\boldsymbol{K}}$ is dependent on the modified densities.

For an assembly optimization, also the derivatives with respect to the joints' reference point coordinates are needed. Using adjoint sensitivity analysis yields:

$$\frac{dc}{dx_i} = -\tilde{\boldsymbol{u}}^T \frac{d\tilde{\boldsymbol{K}}}{dx_i} \tilde{\boldsymbol{u}} \tag{6.29}$$

The augmented stiffness matrix is affected directly by the joints' positions through the coupling equations and indirectly through a change of the masks used for the modified densities (cf. figure 6.5):

$$\frac{d\tilde{\boldsymbol{K}}}{dx_i} = \frac{\partial \tilde{\boldsymbol{K}}}{\partial x_i} + \sum_j \frac{\partial \tilde{\boldsymbol{K}}}{\partial \hat{\varrho}_j} \frac{\partial \hat{\varrho}_j}{\partial x_i} \tag{6.30a}$$

$$= \frac{\partial \boldsymbol{G}}{\partial x_i} + \sum_j \frac{\partial \boldsymbol{K}_m}{\partial \hat{\varrho}_j} \frac{\partial \hat{\varrho}_j}{\partial x_i} \tag{6.30b}$$

Inserting (6.30b) into equation (6.29) and using (6.28) yields:

$$\frac{dc}{dx_i} = -\tilde{\boldsymbol{u}}^T \frac{\partial \boldsymbol{G}}{\partial x_i} \tilde{\boldsymbol{u}} + \sum_j \frac{\partial c}{\partial \hat{\varrho}_j} \frac{\partial \hat{\varrho}_j}{\partial x_i} \tag{6.31}$$

The derivatives $\frac{\partial \hat{\varrho}_j}{\partial (\cdot)}$ used in equations (6.27) and (6.31) are dependent on the type of the joints' material NDS zones and are listed in Appendix E.3 – E.5.

6.6.2 Volume constraints

Volume constraints can be formulated for the global model or on a per-part basis (or even on both scales). When limiting the global volume of the assembly to a maximum global volume fraction k_g the constraint function states:

$$h = \frac{V}{V_0} - k_g \leq 0 \tag{6.32a}$$

$$= \frac{\sum_i \hat{\varrho}_i v_i}{\sum_i v_i} - k_g \leq 0 \tag{6.32b}$$

[19] How to obtain the derivatives of problems involving modified densities is given in Appendix D.7. While the context there is TO involving damage scenarios, the only difference for an assembly optimization concerns the partial derivatives $\frac{\partial \hat{\varrho}_j}{\partial \bar{\varrho}_j}$ that depend on the type of NDS used for the joints and are given in Appendix E.3 – E.5.

Here, v_i is the volume of element i and the total volume is calculated based on the modified densities $\hat{\boldsymbol{\varrho}}$, including all joints' NDS zones.

The volume constraint can also be applied on a per-part basis to have control over how the material is distributed among the parts. The evaluation of the volumes is then performed over a set $\mathbb{L}_j$ containing the element indices of part j only. The maximum volume fraction per part can then be limited to a value k_p:

$$h_j = \frac{\sum_{i\in\mathbb{L}_j} \hat{\varrho}_i v_i}{\sum_{i\in\mathbb{L}_j} v_i} - k_p \leq 0, \qquad j = 1, \ldots, n_p \tag{6.33}$$

The derivatives with respect to the material and position design variables are given in Appendix E.6. In the examples shown in sections 6.8 and 7.2, part-wise volume constraints are applied.

6.6.3 Minimum distance constraint

A minimum distance between individual joints may be needed to avoid overlapping and to ensure a minimum clearance of the fastener elements in the assembly. The presented minimum distance constraint acts on the position DVs $\boldsymbol{x}$ only.

The squared distance s^{ij} between the reference points of joints i and j is:

$$s^{ij} = {d^{ij}}^2 = (\boldsymbol{x}^j - \boldsymbol{x}^i)^T(\boldsymbol{x}^j - \boldsymbol{x}^i) \tag{6.34}$$

The pairwise distances $d^{ij} = \sqrt{s^{ij}}$ should not fall below a desired value d_0. Either each distance may be constrained by an own constraint function or a single aggregated function constraining the minimal distance can be used:

$$h(\boldsymbol{x}) = d_0 - \sqrt{\min_{i\neq j} s^{ij}} \leq 0 \tag{6.35}$$

For gradient-based optimization, a differentiable approximation for the min-operator in equation (6.35) is needed. Common choices for approximating the max-operator are given in section 3. In order to transform the min-problem to a max-problem, a reciprocal mapping (see e.g. [114]) is applied to the squared distances s^{ij}. This ensures, that positive values stay positive, such that aggregation via the p-norm aggregation function A_{pn} (see section 3.2.1) can be performed. After aggregation, the end result needs to be inverted again:

$$h_{agg}(\boldsymbol{x}) = d_0 - \sqrt{\frac{1}{A_{pn}\left((s^{ij}+\epsilon)^{-1}\right)}} \tag{6.36a}$$

$$= d_0 - \left(\sum_i \sum_{j<i} \left(s^{ij}+\epsilon\right)^{-p}\right)^{-\frac{1}{2p}} \leq 0 \tag{6.36b}$$

Equation (6.36) takes into account, that $s^{ij} = s^{ji}$, such that the number of summands is halved and therefore the quality of the approximation is improved. Also, a parameter ϵ with a small positive value is introduced to counter the singularity that would otherwise exist, if points coincide. In this work, a value of 0.01 times the element edge length is used for ϵ and the aggregation parameter p is set to a value of 8. The gradient of the aggregated constraint function is obtained by direct differentiation and is given in Appendix E.7.

6.6.4 Maximum joint force constraint

Constraints on the maximum joint force may be needed either to model maximum allowed forces for the physical fasteners or as a means to suppress designs, in which only a few joints carry most of the load, while other joints are barely loaded or not loaded at all. For the latter case, a maximum force constraint can be used to obtain designs, in which forces are distributed more evenly across several joints.

The forces transferred by a joint are obtained by summing up the individual forces acting on the inserted slave nodes for all springs of the corresponding spring pattern in one layer (cf. figures 6.9 and 6.10). The total force vector $\boldsymbol{f}^i_{inner}$ for all inner spring forces of joint i can be calculated with the corresponding slave node displacements $\boldsymbol{u}^i_c$:

$$\boldsymbol{f}^i_{inner} = \boldsymbol{K}^i_c \boldsymbol{u}^i_c, \qquad i = 1, \ldots, n_j \tag{6.37}$$

Since the slave nodes for the springs are held in place by coupling equations (see section 6.5.3) the forces can also be obtained via the Lagrange multipliers $\boldsymbol{\lambda}^i_{\boldsymbol{c}}$ for these coupling equations.

According to equation (6.19), the coefficients for the slave displacements $\boldsymbol{u}^i_c$ are all -1 for the unscaled coupling equation. Considering equations (6.5a) – (6.8), the corresponding rows of the system equation reveal, that the forces sought are identical to the Lagrange multipliers, if no scaling of the coupling equation is applied[20] and no additional external forces act on these nodes:

$$\boldsymbol{K}^i_c \boldsymbol{u}^i_c - \boldsymbol{\lambda}^i_{\boldsymbol{c}} = \boldsymbol{0} \tag{6.38a}$$

$$\boldsymbol{f}^i_{inner} = \boldsymbol{\lambda}^i_{\boldsymbol{c}} \tag{6.38b}$$

From the force vector $\boldsymbol{f}^i_{inner}$ only the subset of forces acting on one layer (i.e. on one part) are of interest. The forces on the corresponding other connected part are of the same magnitude but act in the opposite direction since the total set of forces in $\boldsymbol{f}^i_{inner}$ are inner forces on the springs that cancel out in static equilibrium. If only the magnitude of the joint force is to be constrained, the direction does not matter. Hence, the subset of forces for any of the two layers, between which the springs act, can be chosen.

The forces on one layer are summed up for each dimension to obtain the resultant joint force $\boldsymbol{F}^i$ (here shown for a 3D example):

$$\boldsymbol{F}^i = \begin{bmatrix} F^i_x \\ F^i_y \\ F^i_z \end{bmatrix}, \quad F^i_x = \sum \lambda^{i,Layer}_{c,x}, \quad F^i_y = \sum \lambda^{i,Layer}_{c,y}, \quad F^i_z = \sum \lambda^{i,Layer}_{c,z}, \quad \forall i \tag{6.39}$$

As shown in the above equation (6.39), the components of the joint force vector $\boldsymbol{F}^i$ are just sums of certain entries of $\boldsymbol{\lambda}^i_{\boldsymbol{c}}$, which are a part of the augmented displacement vector $\tilde{\boldsymbol{u}}$ (see equation (6.7)). The sums can be rewritten as dot products, involving $\tilde{\boldsymbol{u}}$ and selection vectors $\boldsymbol{l}^i$, which have a value of 1 at the λ^i_c positions of interest and are 0 everywhere else:

$$F^i_x = {\boldsymbol{l}^i_x}^T \tilde{\boldsymbol{u}}, \quad F^i_y = {\boldsymbol{l}^i_y}^T \tilde{\boldsymbol{u}}, \quad F^i_z = {\boldsymbol{l}^i_z}^T \tilde{\boldsymbol{u}} \tag{6.40}$$

[20] In cases, where the coupling equations are scaled, a simple reverse scaling recovers the real forces from the Lagrange multipliers.

The normalized maximum force constraint function h_i limits the magnitude of $\boldsymbol{F}^i$ to an upper threshold $\bar{F}_t$:

$$h_i = \frac{\sqrt{\boldsymbol{F}^{iT}\boldsymbol{F}^i}}{\bar{F}_t} - 1 \leq 0, \qquad i = 1, \ldots, n_j \tag{6.41a}$$

$$= \frac{\sqrt{F_x^{i2} + F_y^{i2} + F_z^{i2}}}{\bar{F}_t} - 1 \leq 0 \tag{6.41b}$$

The gradients for each of the n_j force constraint functions h_i can be obtained by adjoint sensitivity analysis (see Appendix E.8). However, this requires one solution of the system matrix per joint for the corresponding adjoint system.

In order to cut down the computational cost, constraint aggregation discussed in section 3.1.2 can be performed to replace the n_j force constraint functions h_i from the equations (6.41) by a single constraint function h_{agg}. The p-norm aggregation function A_{pn} from section 3.2.1 is a suitable choice for a conservative approximation of the maximum force magnitude ratio terms:

$$h_{agg} = A_{pn}\left(\left[\frac{\sqrt{\boldsymbol{F}^{iT}\boldsymbol{F}^i}}{\bar{F}_t}\right]_{i=1,\ldots,n_j}\right) - 1 \leq 0 \tag{6.42}$$

To achieve high accuracy, an aggregation parameter $p = 32$ is used for the p-norm aggregation function.

The advantage of using constraint aggregation here is, that now a single combined pseudo load vector for the adjoint sensitivity analysis can be composed, such that also only one solution of the system matrix is required in order to obtain the gradient of the aggregated maximum force constraint h_{agg}, which is explained in more detail in Appendix E.8.

6.7 Assembly optimization considering failure of joints

To optimize the performance of an assembly for the case, that one or more joints fail, failure cases are defined. In each failure case, the stiffness of a unique combination of joints is degraded by multiplying the corresponding joint stiffnesses with a very low value[21], while all other joints remain intact with their full stiffness. Then, the maximum (worst-case) objective value for all failure cases is minimized.

In contrast to the optimization problems considering failure of the continuum structure (see sections 4 and 5), the number of joints in an assembly is usually low, such that even the combinatorial failure of multiple joints at once may be considered at a reasonable computational cost.

The failure mode m is defined by m joints being considered as failed at the same time. Then, the total number n_d of failure combinations is:

$$n_d = \binom{n_j}{m} = \frac{n_j!}{m!(n_j - m)!} \tag{6.43}$$

[21] Here, a factor of 10^{-9} is used, which yields the same ratio of stiffness between an intact and failed joint as the stiffness ratio between a solid and a void element in the TO design space is (see sec. 2.3.4).

The examples shown in section 6.8 have a maximum number of $n_j = 4$ joints, which yields $n_d = 4$ failure cases for failure mode $m = 1$ and $n_d = 6$ failure cases for failure mode $m = 2$.

Adapting the approach from sections 4.1.4 and 5.2.2, the underlying min-max problem for optimizing the worst-case joint failure scenario is approximated by using the KS aggregation function A_{KS} from section 3.2.3. The optimization task is then obtained by using the equations from (6.4) and exchanging (6.4a) by:

$$\min_{\boldsymbol{\varrho},\boldsymbol{x}} \frac{1}{p} \log \left(\sum_{m=0}^{\bar{m}} \sum_{d=0}^{n_d(m)} \mathrm{e}^{p\Phi_d^m(\boldsymbol{\varrho},\boldsymbol{x})} \right) \tag{6.44}$$

Therein, Φ_d^m denotes the objective function evaluated for damage case d of failure mode m (up to the maximum considered mode $\bar{m}$) and p is a parameter for the aggregation function. The number of damage cases n_d is dependent on the chosen failure mode m and $d = 0$ defines the undamaged case in which all joints are intact.

If compliance is the objective function to be minimized, the worst-case compliance of mode m will always exceed all worst-case compliances of the lower failure modes (including failure mode $m = 0$ that only includes the undamaged configuration):

$$\max_d c_d^m > \max_d c_d^{m-1}, \qquad \forall m = 1, \ldots, n_j \tag{6.45}$$

Therefore, only the failure scenarios from the highest studied failure mode $\bar{m}$ need to be evaluated when optimizing for the worst-case compliance and also the undamaged configuration does not need to be considered. Equation (6.44) is thus simplified to:

$$\min_{\boldsymbol{\varrho},\boldsymbol{x}} \frac{1}{p} \log \left(\sum_{d=1}^{n_d(\bar{m})} \mathrm{e}^{pc_d^{\bar{m}}(\boldsymbol{\varrho},\boldsymbol{x})} \right) \tag{6.46}$$

When considering failure of joints, employing the minimum distance constraint from section 6.6.3 is strongly advised. Otherwise, it is likely, that joints overlap in groups of $m + 1$ joints (meaning e.g. in pairs of two joints, when considering damage of single joints) to achieve a perfect redundancy of the connection which however is unphysical.

6.8 Numerical examples for assembly optimization

The proposed method for assembly optimization is applied to two examples in 2D and one example in 3D.

For the 2D examples, a cantilever beam and an L-shaped bracket are optimized for minimum compliance under part-wise volume constraints. Both structures are optimized as assemblies, consisting of two separate parts connected by joints. For each of the two structures, the compliance of the optimized assemblies is compared to the single-part designs from section 2.7 that serve as reference designs.

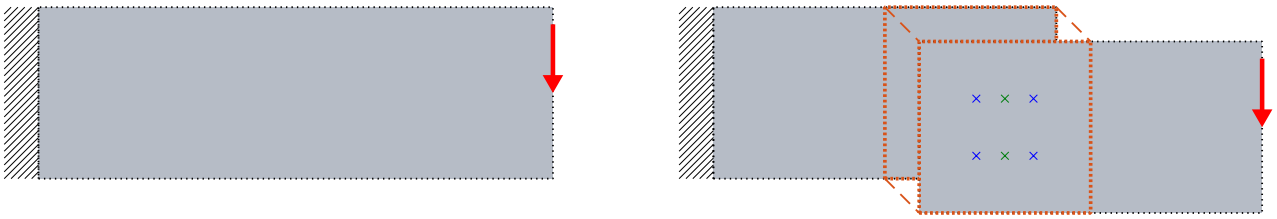
In the 3D example, a plate and a bracket connected by four joints and subjected to a pulling load are studied. The objective is again a minimization of compliance under volume constraints.

The general optimization parameters that are in common for the 2D and 3D examples are listed in Appendix C.6. The joints' resultant stiffness is set to $k_c = 10$ for all examples shown. Problem-specific parameters are given in the corresponding following sections.

6.8.1 2D cantilever beam structure

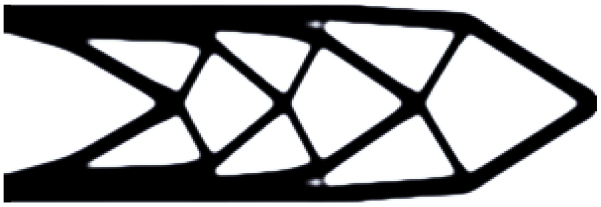
The design domains, boundary conditions, and loads for the 2D cantilever beam example are shown in figure 6.11. The design space of the reference design is discretized by 300×100 finite elements. The two-part design has the same outer dimensions, while each part has a design domain of 200×100 elements. Therefore, an overlapping area of 100×100 elements is present, which bounds the space where the joints can be located.

The initial positions of the joints' reference points for the examples involving two joints are shown in green in figure 6.11b. The positions marked in blue refer to the initial positions for the examples involving four joints. The optimizations are performed with a filter radius of $r = 4$, as in the plain TO examples of section 2.7.



(a) Single-part model

(b) Two-part model with area of overlap



(c) Reference design from fig. 2.8b

Figure 6.11 – Design domains for the 2D cantilever beam examples with boundary conditions, loads, and initial locations of the joints (green, blue) together with the reference design.

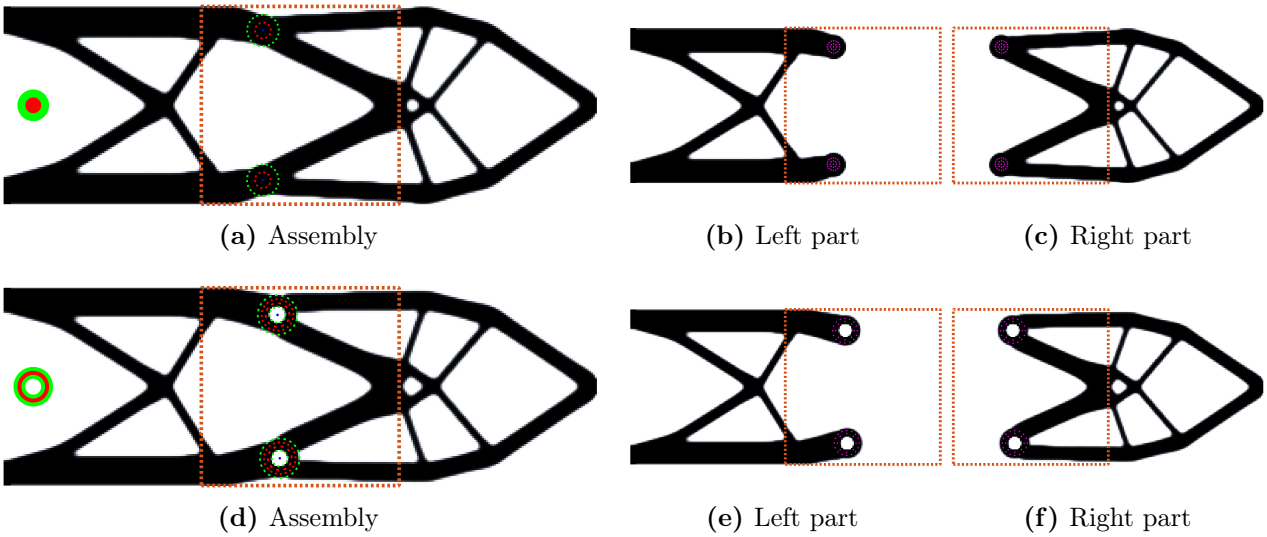
One-part reference design The reference design of compliance minimization with a global volume constraint with $k_g = 0.4$, optimized by the plain TO method from section 2.7.1, is repeated in figure 6.11c. This single-part design has a compliance of $c = 210.19$.

Design with two joints Two different joint types are investigated here: Spot welds, which only require a circular material zone, and bolts, which additionally require a mounting hole through both mating parts.

The load transfer is modeled by concentric patterns of springs. The resultant stiffness of the spring patterns is the same for both connection types and has the value $k_c = 10$. The spring patterns, together with the material and force transfer zones, are shown in figure 6.10. For the spot welds, a material radius of 8 is used, together with a force transfer zone radius of 4. For the bolted design, the material zone has an outer radius of 10, the hole has a radius of 4. The radii for the springs are 6 and 8.

For the assembled designs, a part-wise volume constraint of $k_p = 0.3$ is employed for both parts. Since the total size of the design space is enlarged due to the overlapping region, a value of $k_p = 0.3$ here yields the same allowed global volume as the reference design with $k_g = 0.4$.

The final designs for the spot welded and bolted variants are shown in figure 6.12. The results share the same topology and differ only slightly. This is because the spot welded and bolted joints are chosen with similar sizes. In both cases, the joints are placed on the main load paths on the thick upper and lower beams of the structure.



(a) Assembly **(b)** Left part **(c)** Right part

(d) Assembly **(e)** Left part **(f)** Right part

Figure 6.12 – Results for the cantilever beam with two spot welds (top) and two bolts (bottom).

Numerical values for the compliance c are given in table 6.1 and reveal that the assembled designs suffer only a small penalty of about 6% and 7% compared to the one-piece reference design. The spot-welded design is slightly stiffer than the bolted design since the latter has a weakening hole in the load path and also the size of the enforced material zone is larger, making it harder to integrate it into the structure.

In table 6.1, the total compliance c is further split into the compliance contribution of the material c_m and of the joints c_c, according to equation (6.26). With the parameters used here, the joint compliance is just a very small fraction of the total compliance, such that the optimization problem is dominated by the material distribution and the joint locations, and not by the modeling of the force transfer in the joints.

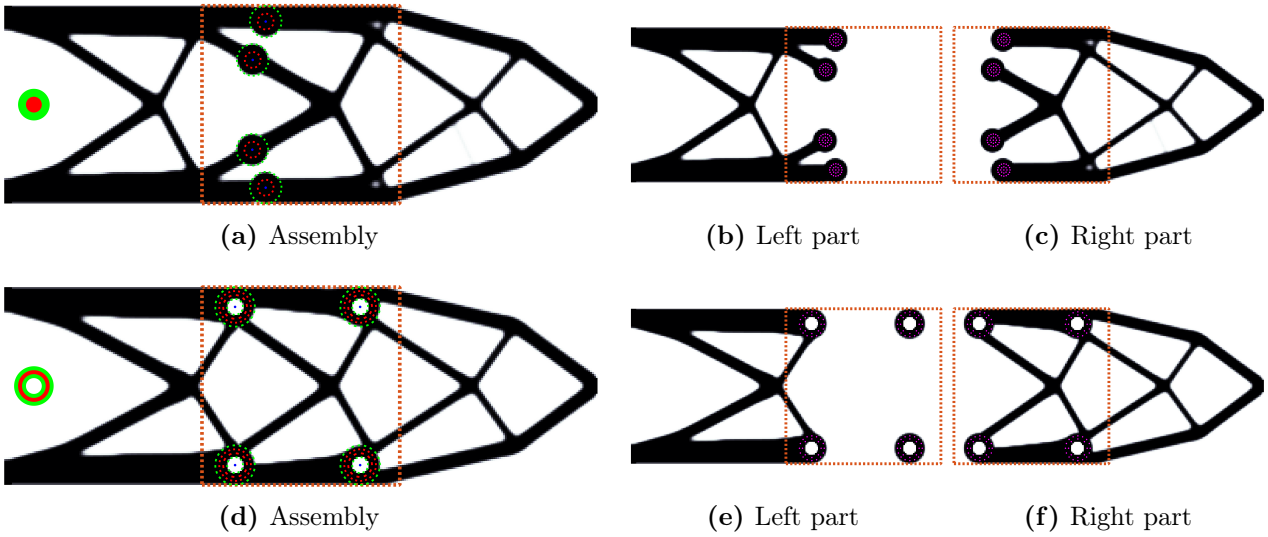
Table 6.1 – Compliance values for the cantilever beam designs with two joints.

Configuration	Figure	Total c	Material c_m	Joint c_c
Reference	6.11c	210.19	210.19	0
2 spot welds	6.12a	223.00	221.80	1.20
2 bolts	6.12d	225.41	224.24	1.18

Design with four joints When increasing the number of joints, the minimum distance constraint from equation (6.36) needs to be included in the optimization task. Otherwise, joints may coincide, which leads to an unphysical joint layout and thus should be avoided. Without modifying the optimization task, the results for the four-joint designs look the same as the results from figure 6.12 with just two joints.

Since the two-joint designs are already close to the single-part reference design in terms of performance, additional joints are virtually unnecessary. When employing a minimum distance

constraint with $d_0 = 20$ (which is the diameter of the NDS for the bolts), the results from figure 6.13 are obtained. Both designs show a strong similarity with the reference design in figure 6.11c. The minimum distance constraint is inactive in both final designs.



(a) Assembly **(b)** Left part **(c)** Right part

(d) Assembly **(e)** Left part **(f)** Right part

Figure 6.13 – Results for the cantilever beam with four spot welds (top) and four bolts (bottom) with minimum distance constraint.

When examining the bolted design from figure 6.13d first, it is evident, that the two rightmost joints do not transfer any load between the parts. The unused joints are placed at a location where the waste of material is minimal. The holes of the corresponding joints' NDS are at the same locations, where the reference design in figure 6.11c also has small holes.

The design with four spot welds from figure 6.13a has smaller NDS material zones for the joints compared to the bolted design, hence the additional joints can be placed more flexibly and are incorporated into two other load paths.

Consequently, since the bolted designs pose a higher challenge for the optimization due to their larger NDS material radius and the central holes, the following optimizations focus on bolted connections.

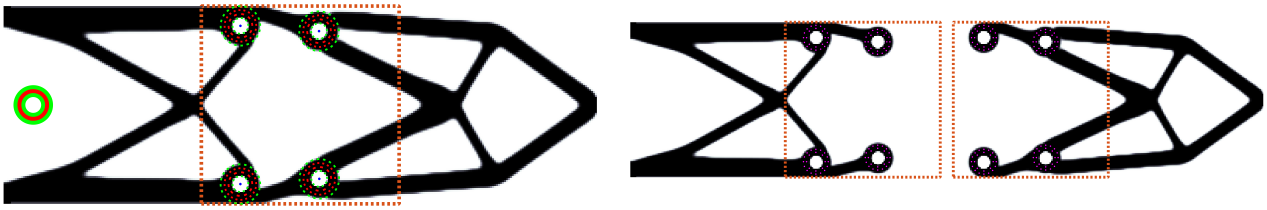
The shown results underline, that a simple increase of the number of joints in a plain assembly optimization does not guarantee, that all joints are placed on different load paths nor that they even transfer load at all. Nevertheless, all additional "excess" joints come with a cost in material through the NDS material zones, therefore the structural performances are worse for the four-joint designs, compared to the two-joint designs, as summarized in table 6.2.

Table 6.2 – Compliance values for the cantilever beam designs with four joints compared to the two-joint designs.

Configuration	Figure	Total c	Material c_m	Joint c_c
Reference	6.11c	210.19	210.19	0
2 spot welds	6.12a	223.00	221.80	1.20
4 spot welds	6.13a	227.36	226.44	0.92
2 bolts	6.12d	225.41	224.24	1.18
4 bolts	6.13d	226.63	224.90	1.73
4 bolts, max. force	6.14a	237.24	236.11	1.13

Design with four joints and a limit on joint forces As a means to counter the unconnected joints in the bolted design from figure 6.13d, the maximum joint force constraint from section 6.6.4 can be employed. Choosing a value for the maximum force $\bar{F}_t$ equal to the magnitude of the applied force at the tip means, that for the given geometry and restrictions for the joint locations, a pair of two bolts is just not capable of absorbing the resulting inner moment. Therefore, in order to yield a feasible design, the interface forces need to be distributed over all four bolts.

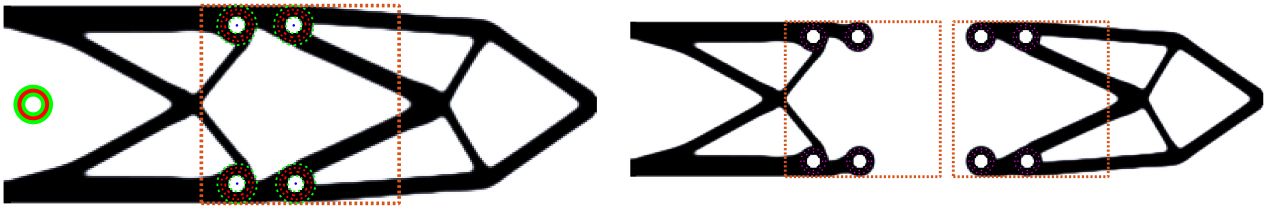
Figure 6.14 shows the resulting design for the four bolt cantilever when additionally to the minimum distance constraint also the maximum force constraint is considered in the optimization. The force is distributed almost uniformly between the four bolts, with values of about 95% and 96% of the maximum allowed force[22]. However, as listed in table 6.2, the compliance is larger than for the four-bolt design from figure 6.13d, which has two unconnected joints but no limitation on the maximum joint forces.



(a) Assembly **(b)** Left part **(c)** Right part

Figure 6.14 – Results for the cantilever beam with four bolts, minimum distance, and maximum force constraint.

Design with four joints considering joint failure When considering the failure of single joints, it is even more important to employ a minimum distance constraint, otherwise, joints will overlap in pairs of two for ideal (but unphysical) redundancy. When enforcing a minimum distance, the joints are pushed further apart. For the bolted design and $d_0 = 20$, the result in figure 6.15 is obtained, which also has no unused joints and is almost identical to the design in figure 6.14. The visible separation of the joints' NDS reveals that the minimum distance constraint is again not active in the final result.



(a) Assembly **(b)** Left part **(c)** Right part

Figure 6.15 – Results for the cantilever beam with four bolts and minimum distance constraint, fail-safe joint design.

Table 6.3 lists the compliance values for the nominal, undamaged case as well as for the worst-case failure of a single joint ($\bar{c}^1$ for failure mode $m = 1$) for all shown four joint designs.

[22]Due to the conservative p-norm aggregation used in the maximum force constraint (see section 6.6.4), the real maximum is overestimated and the joint forces could not go beyond the given values.

Table 6.3 – Compliance values for the cantilever beam designs with four joints and single joint failure.

Configuration	Figure	$\bar{c}^1$	c_0
4 spot welds	6.13a	1 665.92	227.36
4 bolts	6.13d	7 903.42	226.63
4 bolts, max. force	6.14a	249.01	237.24
4 bolts, fail-safe joints	6.15a	239.29	231.43

Considering a single joint failure, the four-bolt design from a plain assembly optimization shown in figure 6.13d is in the worst case left with only one load-bearing joint, causing a severe drop of stiffness. Even though the spot-welded design from figure 6.13a has its four joints placed on different load paths, still a significant increase in compliance is observed, if one of the outer spot welds fails.

In contrast to that, the four-bolt design from figure 6.14a suffers only from a small penalty on the overall stiffness, if one joint fails. The increased damage tolerance is achieved in this case implicitly by limiting the maximum joint force to a suitable value and thus distributing the interface loads evenly across all joints. Still, no explicit evaluation of the damage cases is performed during the optimization. As a result, the maximum allowed joint forces are exceeded if one joint fails, which may not be critical if the maximum force constraint is only used as a means to avoid unconnected joints and the value of $\bar{F}_t$ does not correspond to a real maximum allowed force for the physical fastener.

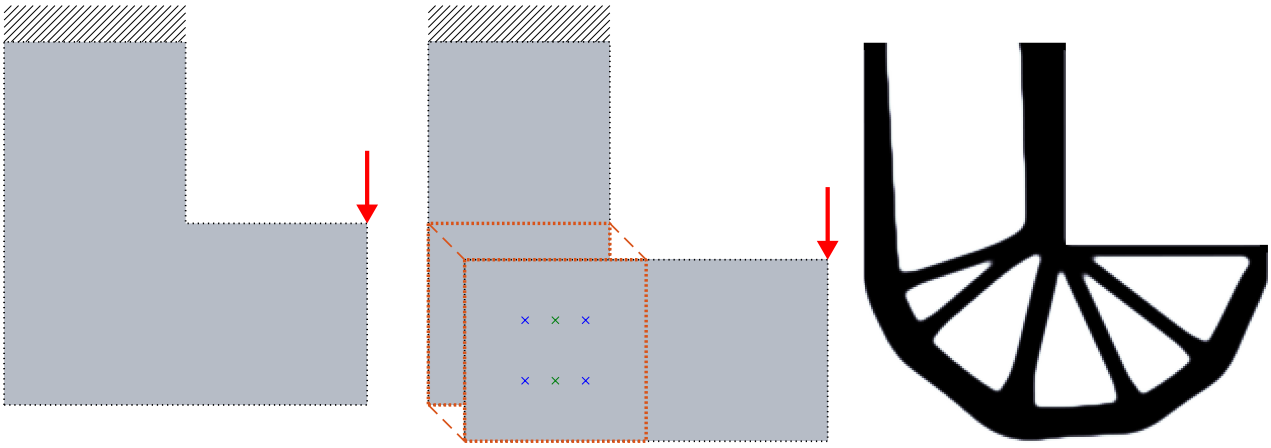
The four-bolt design with a fail-safe joint layout from figure 6.15a is explicitly optimized for the damaged configuration and thus has the lowest worst-case compliance under joint failure, as listed in table 6.3. However, this approach requires n_d FE analyses per optimization iteration (cf. equation (6.43)). The obtained damage tolerant design is in this case also stiffer for the undamaged configuration, compared to the four-bolt design with a limit on the maximum joint forces from figure 6.14a. Nevertheless, for the undamaged case, both designs are more compliant than the design from figure 6.13d, even though the latter has unused joints.

6.8.2 2D L-bracket structure

Figure 6.16 shows the design domains, boundary conditions, and loads for the 2D L-bracket example. The outer dimensions of the design space for the reference design are 200×200 finite elements, with the upper right 100×100 elements left out, forcing the structure to go around a corner. The assembled design consists of two distinct design spaces with dimensions of 100×200 and 200×100 elements, respectively. The overlapping region, in which the joints are located, has again a size of 100×100 elements, just like in the cantilever beam example from section 6.8.1.

Bolted joints with the same NDS sizes as for the cantilever beam example are considered for the L-bracket examples. The initial positions of the joints' reference points for two joints (green) and four joints (blue) are shown in figure 6.16b. Again, a filter radius of $r = 4$ is used for the optimizations.

One-part reference design The design from section 2.7.2 serves as one-piece reference design and is repeated in figure 6.16c. The structure is obtained by compliance minimization with a global volume constraint with $k_g = 0.4$, which yields a final compliance of $c = 100.29$.



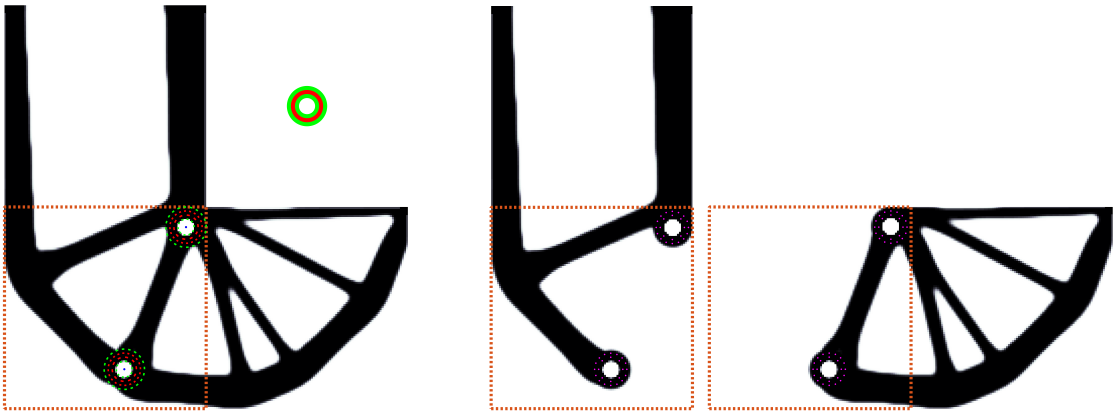
(a) Single-part model **(b)** Two-part model with area of overlap **(c)** Reference design from fig. 2.9b

Figure 6.16 – Design domains for the 2D L-bracket examples with boundary conditions, loads, and initial locations of the joints (green, blue) together with the reference design.

Design with two joints A part-wise volume constraint of $k_p = 0.3$ is employed for both parts of the assembly, resulting in the same allowed global volume as for the one-part reference design.

The optimized design is shown in figure 6.17 and is very similar to the reference design. With a compliance value of $c = 108.83$, the assembled design suffers a penalty of less than 9% when compared to the reference design.

The joints are placed far apart to efficiently absorb the inner bending moment introduced by the external tip load. The upper joint is shifted as far into the corner as possible. Furthermore, both joints are located at intersections of structural features in the density field, such that the weakening effect of the central hole of the joints' NDS is minimal.

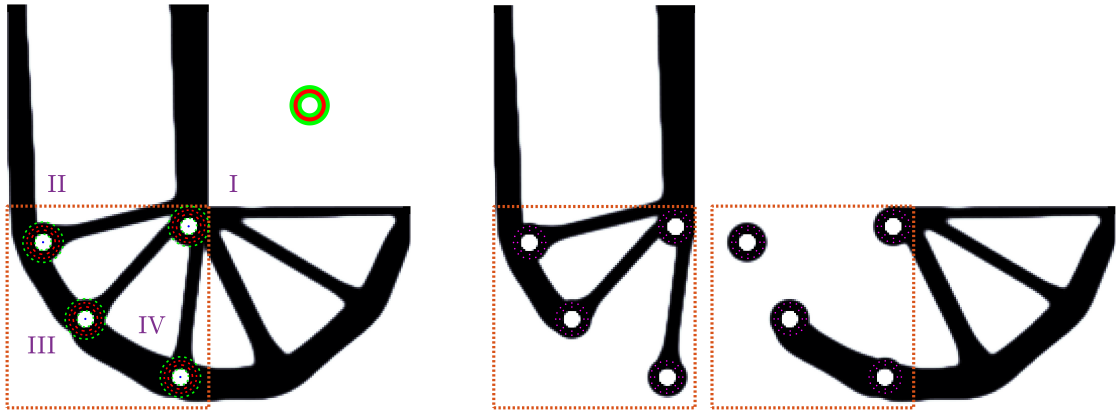


(a) Assembly **(b)** Left part **(c)** Right part

Figure 6.17 – Results for the L-bracket with two bolts.

Design with four joints Due to the increased number of joints, a minimum distance constraint with $d_0 = 20$ is employed to prohibit optimization results with overlapping joints.

The obtained final result is shown in figure 6.18, the corresponding compliance value is $c = 113.21$. Similar to the cantilever beam examples from section 6.8.1, the compliance of the four-joint L-bracket design is higher than for the two-joint design, due to one unused and one "inefficient" excess joint. Table 6.4 summarizes the compliance values, as well as their material and joint contributions, of the reference, two-bolt, and four-bolt L-bracket designs.



(a) Assembly **(b)** Left part **(c)** Right part

Figure 6.18 – Results for the L-bracket with four bolts and minimum distance constraint.

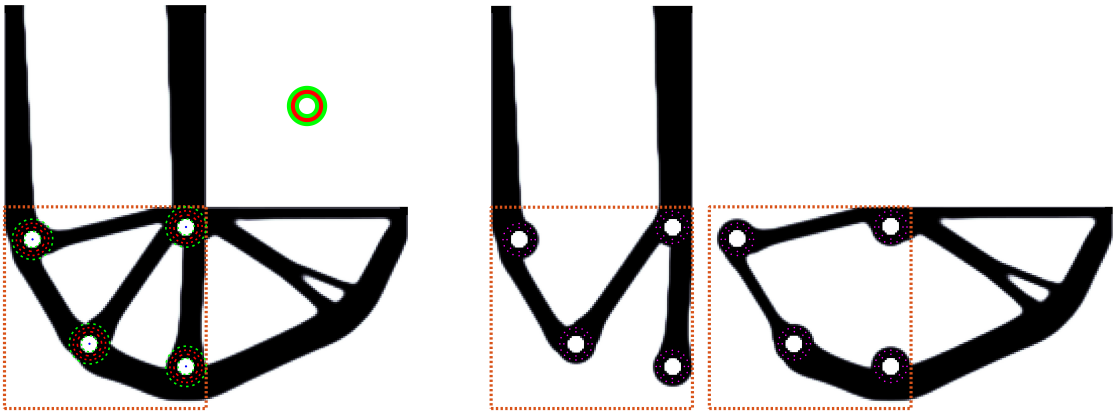
Compared to the result with two bolts from figure 6.17, it can be seen, that again one joint (joint "I" in figure 6.18a) is located directly at the corner of the L-bracket. Since there is only room for one joint to be at the corner, the three other joints are placed on the arc-like segments opposite to joint "I". Again, all joints are located at intersections of structural features in the density field.

Joint "II" is not connected to any load-bearing member of the right part. It is placed at a position where the waste of material of the NDS and the weakening effect of its hole are minimal. Joint "IV" is not efficient for loads in horizontal direction, due to its connection to an almost vertical slender beam on the left part. The main inner bending moments are therefore carried by joint "I" in conjunction with joint "III", similar to the two-joint design.

Table 6.4 – Compliance values for the bolted L-bracket designs.

Configuration	Figure	Total c	Material c_m	Joint c_c
Reference	6.16c	100.29	100.29	0
2 bolts	6.17a	108.83	107.65	1.19
4 bolts	6.18a	113.21	111.72	1.49
4 bolts, max. force	6.19a	113.98	113.00	0.98

Design with four joints and a limit on joint forces By again adding the maximum force constraint from section 6.6.4 to the optimization, with a value for $\bar{F}_t$ equal to the magnitude of the applied force, the result from figure 6.19 is obtained. Therein, the force is distributed more evenly across all four joints and no unconnected joints are present. The compliance is slightly higher than for the design without the joint force constraint from figure 6.18 (see table 6.4).

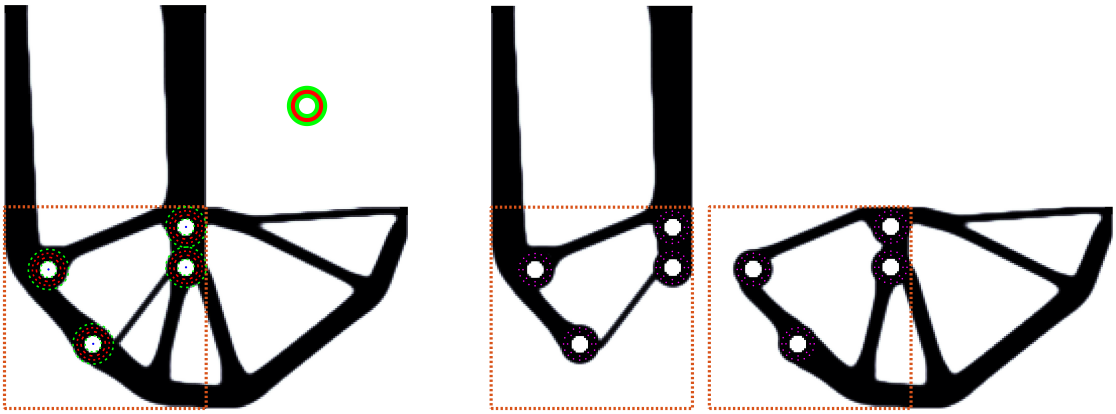


(a) Assembly **(b)** Left part **(c)** Right part

Figure 6.19 – Results for the L-bracket with four bolts, minimum distance, and maximum force constraint.

Design with four joints considering joint failure In this optimization, the failure of a single joint is considered. A minimum distance constraint with $d_0 = 20$ is used. This constraint hinders, that the material zones of neighboring joints overlap.

The optimization yields the result shown in figure 6.20, which has no unused joints. The joints are now grouped in pairs of two to offer robustness towards the failure of a single joint. One joint is again located directly in the corner of the L-bracket, another joint is placed closely below it, such that the minimum joint separation distance is just fulfilled. The other two joints are placed opposite to the L-bracket's corner, such that for every single joint failure always at least two joints remain that have a large lever arm to absorb the inner bending moment efficiently.



(a) Assembly **(b)** Left part **(c)** Right part

Figure 6.20 – Results for the L-bracket with four bolts and minimum distance constraint, fail-safe joint design.

The compliance values for the nominal, undamaged case as well as for the worst-case failure of a single joint are given in table 6.5. When compared to the four-joint design from figure 6.18, the nominal compliance of the design with the fail-safe joint layout is slightly higher, while the performance in the worst case is significantly better.

For the design of a plain assembly optimization shown in figure 6.18, a failure of joint "III" represents the worst-case scenario. In this case, high horizontal loads act on joint "IV", which is

Table 6.5 – Compliance values for the L-bracket designs with four bolts and single joint failure.

Configuration	Figure	$\bar{c}^1$	c_0
4 bolts	6.18a	2 050.62	113.21
4 bolts, max. force	6.19a	254.52	113.98
4 bolts, fail-safe joints	6.20a	127.40	116.90

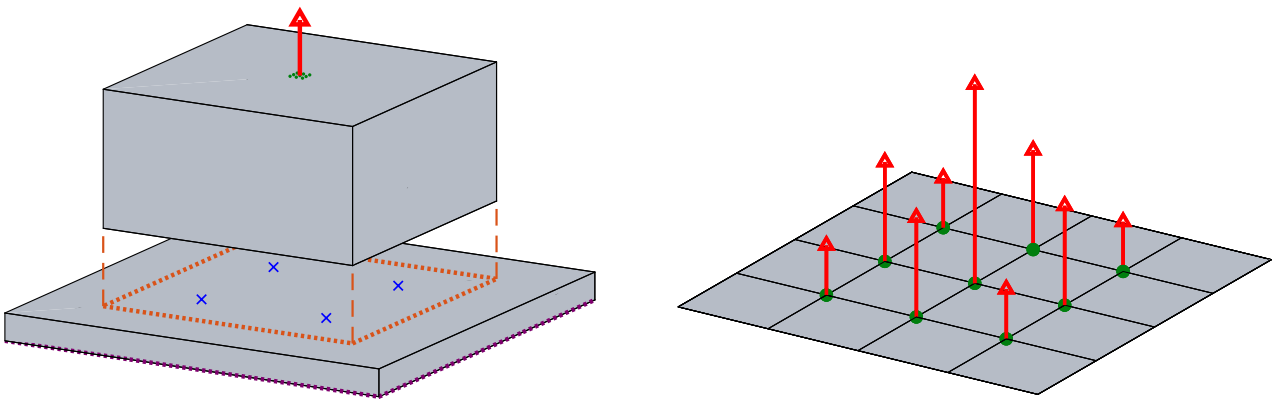
only sub-optimally supported by the left part for this loading direction. Hence, the stiffness of the assembly is severely degraded for this damage case.

The design from figure 6.19 makes use of all four joints and the effect of a single joint failure is cut down significantly, when compared to the design from figure 6.18, as the compliance values in table 6.5 show. The worst-case damage is the failure of the joint in the corner of the L-bracket.

For the structure with the fail-safe joint layout from figure 6.20, the worst-case scenario is also the failure of the joint in the corner, but since a second joint is just below it, the resulting loss of stiffness is comparably small.

6.8.3 3D bracket on plate

For the 3D example, a plate and a bracket connected by four joints and subjected to a pulling load are studied. The design spaces are shown in figure 6.21a. The plate is fixed along the lower edges in the upward direction and at one corner also in the in-plane directions. The load points upwards and is distributed on the nine central nodes of the upper surface of the bracket, as shown in figure 6.21b.



(a) Contact surface (orange), boundary condition edges (purple) and initial joint positions (blue)

(b) Load distribution at the top

Figure 6.21 – Design space and loading for the 3D examples.

The bracket is discretized by $40 \times 40 \times 20$ finite elements and the plate by $60 \times 60 \times 4$ elements.

A NDS material zone with the shape of a hollow cylinder is used for every joint, as shown in figure 6.22. The outer material radius is 5, the hole has a radius of 2.5. The height of the material zone is 4 on the bracket and also 4 on the plate side. Above the material zone, a clearance zone, granting accessibility for bolts mounted from the top, is considered. The top clearance area only affects the bracket and has an outer radius of 5. The spring pattern for the force transfer zone shown in figure 6.22 is equivalent to the pattern shown in figure 6.10b with radii 3.3 and 4.2, and is located on the contact surface of both parts.

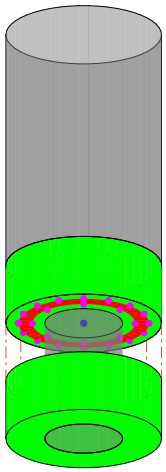


Figure 6.22 – NDS Solid zones (green, red), clearance (gray), reference point (blue), and spring pattern (magenta) used in the 3D examples.

While the bracket undergoes a three-dimensional topology optimization, the plate is optimized for an extruded "2.5D" design, with no density variation along its thickness. Part-wise volume constraints are employed with a volume fraction of 15% for the bracket and 40% for the plate.

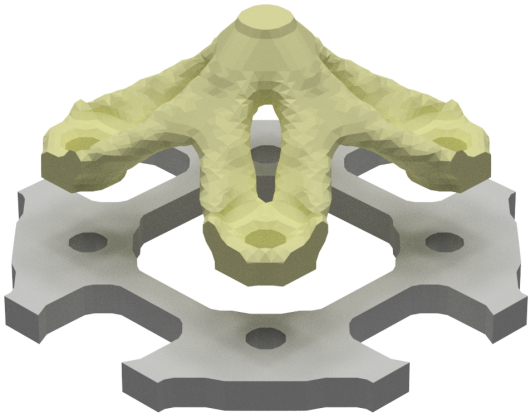
Results for the 3D examples are depicted in figure 6.23. The shown parts are obtained by extracting an isosurface of the density field from the finite element mesh at $\hat{\varrho}_i = 0.5$, yielding a tessellated surface of the parts. The parts are shown with an offset, such that also the plate can be seen.

The result for the optimization without consideration of joint failure is shown in figure 6.23a. Since it is considered that both parts carry the load as an assembly, the mounting points are moved as far out as possible, such that the bracket has the maximum width. In contrast to that, if the bracket was optimized as a single part under a pulling load, a compact rod would be obtained. If mounted on a plate, the plate would then be loaded unfavorably at its most compliant location in the middle. This example demonstrates the importance of optimizing assemblies as a whole, to counter the effect that a single optimized part may introduce suboptimal interface loads to surrounding parts.

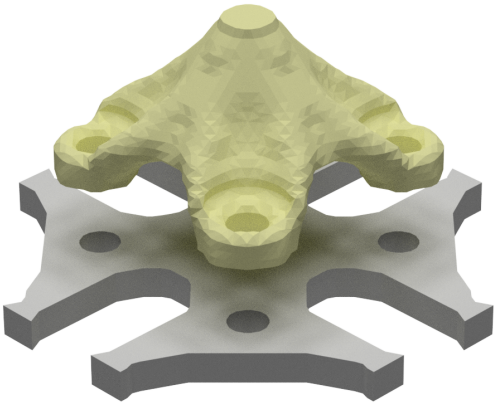
Figures 6.23b and 6.23c show the resulting designs when considering failure of a single joint ($m = 1$) or two joints at once ($m = 2$), respectively. For increased robustness with respect to joint failures, the individual joints are grouped closer together. However, the nominal compliance for the undamaged case suffers an increasing penalty, when comparing the numerical values for the compliance given in table 6.6. The rows in table 6.6 refer to the failure mode considered in the optimization. The columns show the compliance values for the nominal, undamaged case (c_0) as well as the worst-case damages for a single and a double joint failure ($\bar{c}^1$ and $\bar{c}^2$, respectively).

Table 6.6 – Compliance values for the 3D designs with four bolts.

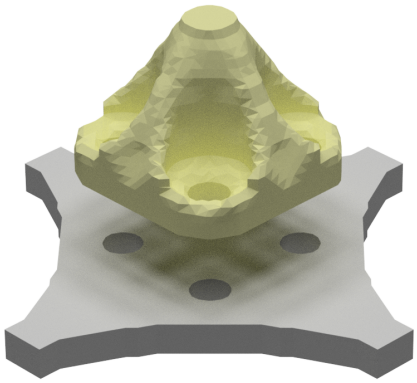
Configuration	Figure	c_0	$\bar{c}^1$	$\bar{c}^2$
Without joint failure	6.23a	2.0893	4.1229	25.0710
Single joint failure	6.23b	2.7796	3.7467	13.4129
Double joint failure	6.23c	4.3491	4.8565	6.9133



(a) Result without considering joint failure



(b) Result considering single joint failure



(c) Result considering double joint failure

Figure 6.23 – Results for the 3D assembly optimization example.

7 Assembly optimization considering fail-safety

Section 6.7 discusses assembly optimization considering failure of joints only. In section 5, also a method for the numerically efficient fail-safe optimization of single components involving failure of load-bearing members is presented. This section is dedicated to the combination of both aspects, namely to the optimization of assemblies considering failure of joints and load paths. The theory is discussed in section 7.1, while numerical examples are given in section 7.2.

7.1 Assembly optimization considering failure of load paths and joints

The proposed methods for assembly optimization and load-path-based fail-safe optimization are both extensions of a standard three field TO and are thus compatible with each other such that a combined consideration is straightforward. Consequently, the two-stage procedure from section 5 is adapted for a use in the context of assembly optimization to efficiently optimize assemblies towards a fail-safe design. Therein, the first optimization stage yields again only a preliminary design that is then shape optimized in a second optimization stage.

In stage I, an assembly optimization following the workflow discussed in the previous parts of section 6 is conducted. Damage on load paths is not considered yet in stage I, instead, local volume constraints are used to enforce parts with redundant load paths as an implicit way to aim for a damage tolerant design (cf. section 5.2). However, it is recommended to already consider failure of joints in stage I. As underlined by the examples shown in section 6.8, considering joint failure yields different joint layouts compared to plain assembly optimizations. Therefore, it is important to take measures such that the joints are already placed in a damage tolerant fashion for the preliminary design, as in stage II a useful relocation of joints might not be possible anymore.

After stage I is finished, a load path identification (cf. section 4.3.4) is performed for all of the n_p parts of the assembly to prepare the load-path-based damage scenarios for the downstream optimization stage.

In the density-based SO of stage II, in addition to joint failure also damage on the previously identified load paths is considered. Just like holes for the joints' NDS are employed by masks in an assembly optimization, damage on load paths is included the same way, i.e. as zones of void defined by a mask. The difference is, that the damage zones are not considered as movable and therefore their masks do not have a derivative with respect to the joints' position DVs.

A simplified variable processing diagram for the two stages is presented in figure 7.1, which combines the workflows shown previously in figures 5.3 and 6.6. Therein, $\boldsymbol{\psi}_J$ denotes the masks used to incorporate the NDS of the joints into the design, while the $\bar{}\boldsymbol{\psi}_i$ are the masks modeling the n_{dL} cuts of individual load paths. The $\boldsymbol{k}_{c,i}$ are vectors storing specific combinations of intact and degraded resultant joint stiffnesses to describe the n_{dJ} cases of joint failure, while $\boldsymbol{k}_{c,0}$ represents the case where all joints are intact. As before, the modified densities $\hat{\boldsymbol{\varrho}}_i$ are used as physical variables for the FE analyses. The modified densities include either the NDS of the joints (denoted as $\hat{\boldsymbol{\varrho}}_0$) or additionally also include a cut for one of the n_{dL} failure cases for load-path-based damage (yielding $\hat{\boldsymbol{\varrho}}_1, \ldots, \hat{\boldsymbol{\varrho}}_{n_{dL}}$). Consequently, three groups of configurations of the numerical model are defined:

1. The configuration with $\boldsymbol{k}_{c,0}$ and $\hat{\boldsymbol{\varrho}}_0$, where all joints and load paths are intact. This configuration can be omitted, if worst-case compliance is the objective.
2. The configurations with $\boldsymbol{k}_{c,1}, \ldots, \boldsymbol{k}_{c,n_{dJ}}$ in conjunction with $\hat{\boldsymbol{\varrho}}_0$, where joint failure is considered but the load paths are still intact.
3. The configurations with $\boldsymbol{k}_{c,0}$ in conjunction with one of the $\hat{\boldsymbol{\varrho}}_1, \ldots, \hat{\boldsymbol{\varrho}}_{n_{dL}}$, where the joints are intact but single load paths are cut.

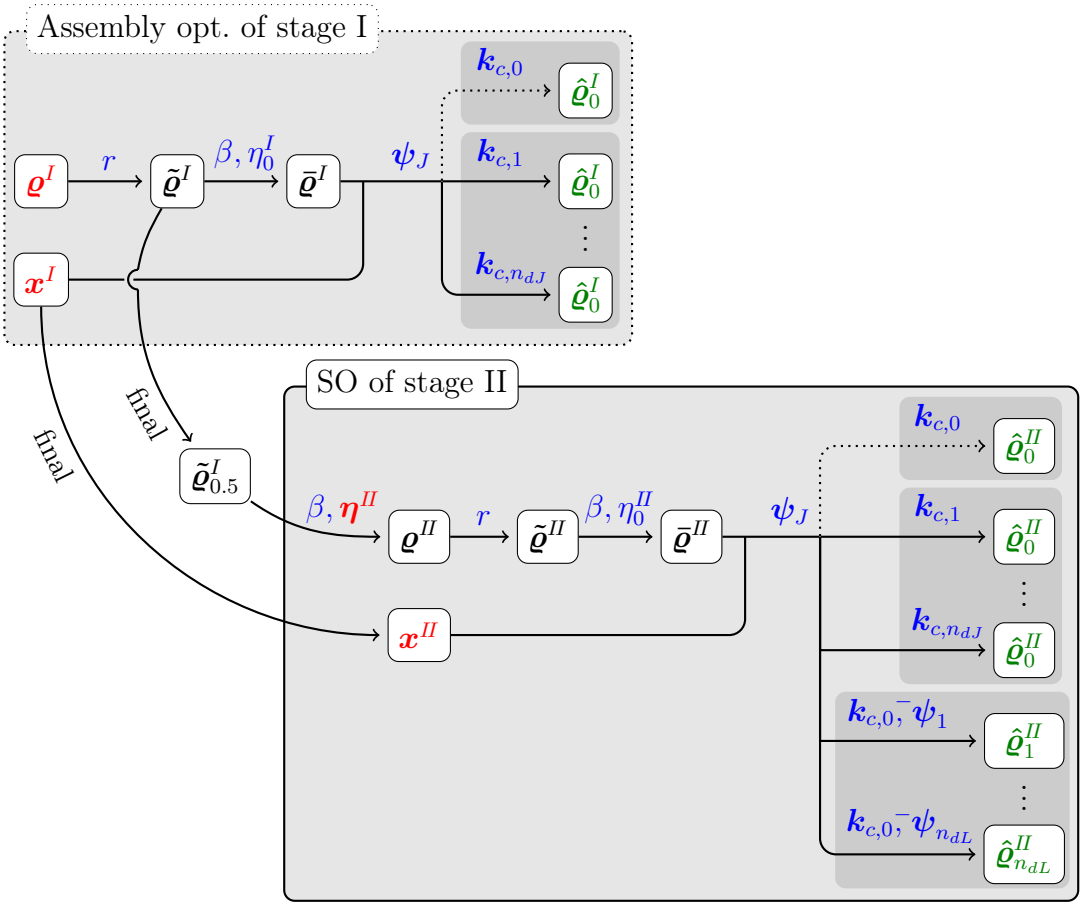


Figure 7.1 – Relation of variables for a two-stage assembly optimization including failure of joints and load paths.

For optimization stage I, only the first two groups of configurations are relevant. In contrast to that, all three groups are relevant and require FE analyses in stage II (see figure 7.1). Another difference is, that in stage II a projection step with the DVs $\boldsymbol{\eta}^{II}$ precedes the material variable processing steps of the assembly optimization, turning this stage from a TO into a density-based SO, as discussed in section 5.

A slight movement of joints is allowed in stage II to adapt to the changed loading conditions due to the additionally considered damage on load paths. It is assumed, that the joints stay on the same load paths as before. Nevertheless, to allow movement of a joint and its corresponding NDS without leaving holes or unconnected load paths at the original positions, the underlying filtered field $\tilde{\boldsymbol{\varrho}}^{I}$ from stage I needs to be slightly modified, before it is used as an input for stage II: The filtered variables under the joints' NDS zones are reset to an intermediate value of e.g. 0.5, in order to allow the projection of stage II to reassign either void or solid material to these regions just by altering the corresponding DV entries in $\boldsymbol{\eta}^{II}$. If however a joint does not move, this modification has no impact on the final design. The altered filtered field is denoted as $\tilde{\boldsymbol{\varrho}}^{I}_{0.5}$ in figure 7.1.

In the context of assembly optimization, damage of joints relates to degrading the stiffness properties of the affected joints for each failure case by altering the corresponding sub-matrices $\boldsymbol{K}_c^i$ of the augmented system matrix $\tilde{\boldsymbol{K}}$ from equation (6.6). Damage on load paths however has an effect on the stiffness matrices $\boldsymbol{K}_m^i$ of $\tilde{\boldsymbol{K}}$, describing the stiffness due to the material distribution of the affected part i.

7.2 Numerical examples for assembly optimization with load path and joint damage

In the following, the 2D cantilever beam and L-bracket examples studied in section 6.8 are optimized towards fail-safe assemblies in considering failure of joints and load paths by using the two-stage optimization procedure from section 5.

The preliminary design for the assemblies is calculated in stage I, where part-wise volume constraints with maximum volume fractions of $k_p = 0.4$ are used (see equation (6.33)). To enforce designs with redundant load paths, aggregated local volume constraints with local volume fractions of $k_l = 0.5$ are employed for each part, using the same control shapes as for the redundant single-part examples in section 5.3. The local volume constraints are applied to the projected variable field, which does not yet include the NDS material zones (cf. equation (2.16)).

Although using above-mentioned settings leads to a higher total volume of the assembly due to the parts' overlap regions compared to the designs shown in section 6.8, which use lower part-wise volume constraints (namely $k_p = 0.3$), these constraints are only used as a means to get the desired degree of redundancy comparable to the examples from section 5.3. However, fitting of the total assembly volume to a specific lower value can be done optionally in stage II.

The subsequent load path identification is done on the embedding structure around the joints' NDS only (i.e. the ring-shaped NDS zones are not identified as load-bearing members unlike everything that connects to it). Wherever possible, the damage case reduction method mentioned in section 5.4 is applied to reduce the number of failure cases involving load paths.

In stage II, failure of joints, modeled by a degradation of the joint spring stiffness, or failure of load paths, modeled by cutting load paths, is considered, one failure at a time. No combined failures, like e.g. a simultaneous failure of one joint and one load path, are considered, although this would also be possible within reasonable computational time using the proposed methods for fail-safe optimization. A slight movement of the joints (within a maximum distance chosen to be the radius of their NDS zones) is allowed in stage II, a minimum distance constraint is employed in cases where this approach results in the risk of overlapping joints.

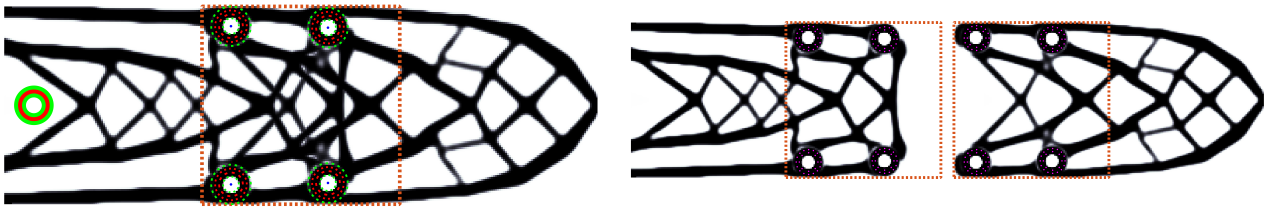
7.2.1 2D cantilever beam structure

The setup of the numerical model for the cantilever beam example is adapted from figure 6.11b and also the corresponding chosen parameters are the same as listed in section 6.8.1. A design with four bolts is studied in the following.

Preliminary design The preliminary design of the assembly optimization considering joint failure with additionally enforced redundancy for both parts is shown in figure 7.2. Due to the already mentioned higher total volume, this design is referred to as "thick" design from stage I.

The locations of the joints are comparable to the four-bolt design considering joint failure but without enforced redundancy from figure 6.15. The overall distribution of load-bearing members is similar to the redundant single-part design shown in figure 5.7a previously used as preliminary design for the proposed two-stage fail-safe optimization procedure. Consequently, the obtained design from figure 7.2 incorporates the four joints in a fail-safe manner into a base design with redundant load paths, that offers increased damage tolerance with respect to failure of single load paths.

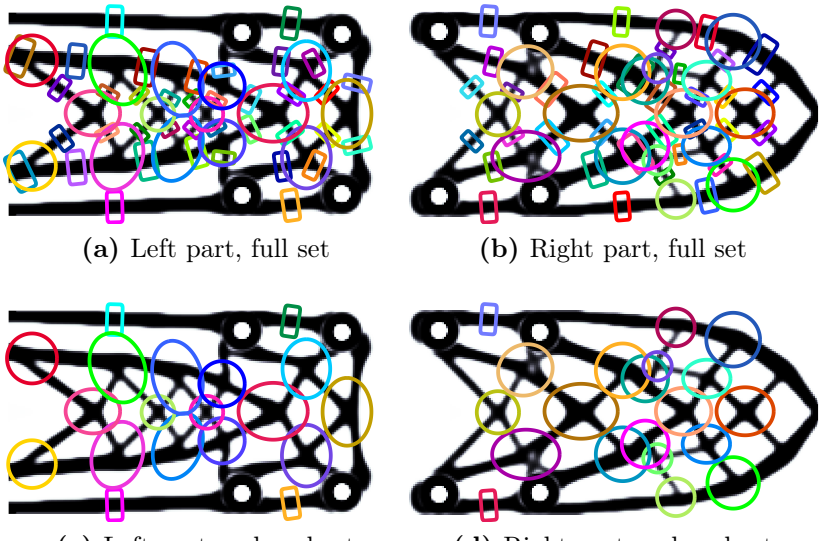
Identified load paths The full set of identified damage shapes for the load paths, obtained by using the image processing technique, is shown in figures 7.3a and 7.3b. In total, 107 load



(a) Assembly **(b)** Left part **(c)** Right part

Figure 7.2 – Results for the redundant "thick" cantilever beam from stage I with four bolts, minimum distance constraint, and consideration of joint failure.

paths are identified (38 beams and 15 knots for the left part, 36 beams and 18 knots for the right part). By applying the damage case reduction technique mentioned in section 5.4, a reduced set of damage shapes with a total count of only 39 shapes is derived, which is shown in figures 7.3c and 7.3d. In this case, some damage scenarios affecting beams cannot be neglected in the course of the reduction, since the corresponding beams connect either two NDS zones or the BC region and a NDS zone in a straight line, without interruption by a knot. The respective four damage shapes for the left part and two shapes for the right part concerning beams are selected manually for the reduced set.



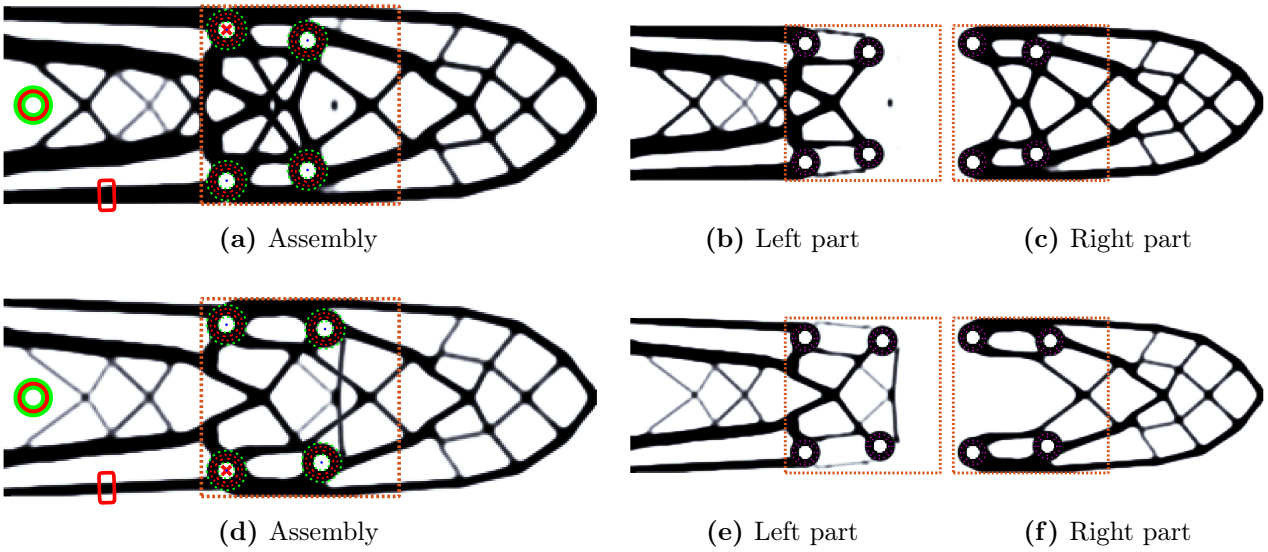
(a) Left part, full set **(b)** Right part, full set

(c) Left part, reduced set **(d)** Right part, reduced set

Figure 7.3 – Identified damage shapes for the redundant cantilever beam from stage I involving the full set of shapes (top) or the reduced set (bottom).

Shape optimized design In the density-based SO of stage II, failure of one of the four joints or one of the 39 critical load paths is considered, while the design with the lowest worst-case compliance is sought.

Two versions of the fail-safe optimizations are conducted: Either the parts' volumes are constrained to not exceed their volumes from the preliminary design of stage I, which yields the shape optimized "thick" design, or each part's volume fraction is reduced to a maximum value of 30%, yielding a "thin" design. Both resulting designs are shown in figure 7.4.



(a) Assembly **(b)** Left part **(c)** Right part

(d) Assembly **(e)** Left part **(f)** Right part

Figure 7.4 – Results for the shape optimized redundant cantilever beam after stage II considering joint failure and damage on load paths: "Thick" design (top) and "thin" design (bottom) with less volume. Worst-case damages are marked in red.

The thick design from figures 7.4a – 7.4c helps to evaluate the effect of the SO of stage II involving additional damage on load paths. The numerical values for the nominal compliance c_0, the worst-case compliance for a single joint failure or a single failure of a load path ($\bar{c}_J^1$ and $\bar{c}_{LP}$, respectively) and the parts' volume fractions ω_p are summarized in table 7.1.

The values reveal, that the thick design from stage II has improved in all aspects compared to the design of stage I since in stage II only part-wise volume constraints are employed and the more restrictive local volume constraint is deactivated. The largest improvement is achieved for the worst-case damage on load paths, which is cut down significantly to about 58% of its original value by the SO of stage II. The worst-case damage locations however stay the same as for the preliminary design and are marked in red in figure 7.4a, where the worst case on the load paths is shown as red shape and the joint with the worst-case joint failure is marked with a red cross.

Table 7.1 – Compliance values for the redundant cantilever beam example considering damage of joints only or damage of joints and load paths.

Configuration	Figure	$\bar{c}_J^1$	$\bar{c}_{LP}$	c_0	ω_p
4 bolts, stage I, thick	7.2a	286.99	675.33	274.52	39.11%, 38.08%
4 bolts, stage II, thick	7.4a	279.72	391.16	234.20	39.09%, 38.08%
4 bolts, stage II, thin	7.4d	528.12	580.86	352.78	29.98%, 29.99%

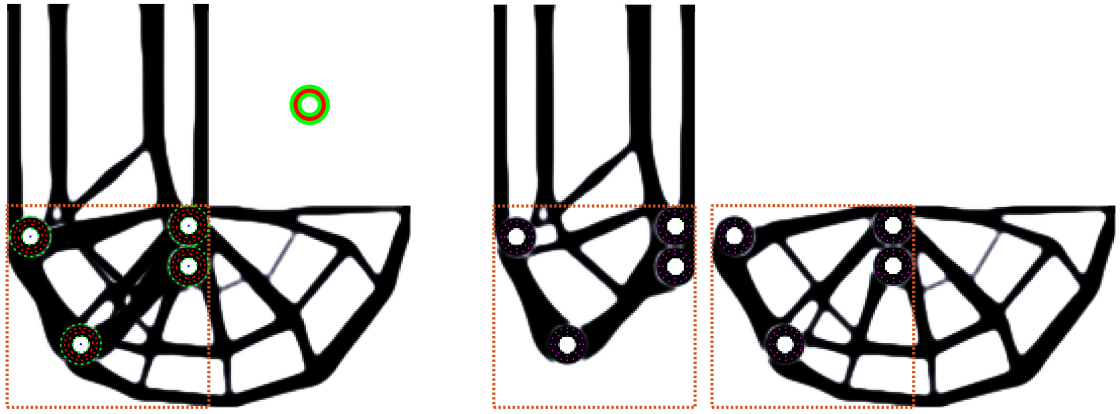
For the thin design with reduced part volumes shown in figures 7.4d – 7.4f, the compliance values are, as expected, increased compared to the optimized thick design. However, the worst-case damage, involving failure of a load path, has still improved (see table 7.1) compared to the initial design from stage I, despite the latter one has more volume. On the downside, the effect of a pure joint failure (denoted as $\bar{c}_J^1$) is increased, since the joint failure cases were not the driving damage scenarios of the optimization. As a result, the material zones around the joints are now only connected by rather thin load paths on the left part (see figure 7.4e).

Interestingly, the performance of the optimized redundant thin two-part cantilever is close to the one of the shape optimized single-part redundant cantilever from figure 5.7c (cf. tables 5.2 and 7.1). In terms of worst-case compliance, the assembled design only suffers a penalty of less than 1%.

7.2.2 2D L-bracket structure

An L-bracket consisting of two parts, connected by four bolts, is studied employing the numerical model shown in figure 6.16b and using the settings given in section 6.8.2.

Preliminary design The redundant preliminary design obtained by optimization stage I, which only considers joint failure, is shown in figure 7.5. The placement of the joints is again qualitatively the same as for the non-redundant design considering single joint failure from figure 6.20. The distribution of structural members is similar to the single-part design with enforced redundancy shown in figure 5.13a.



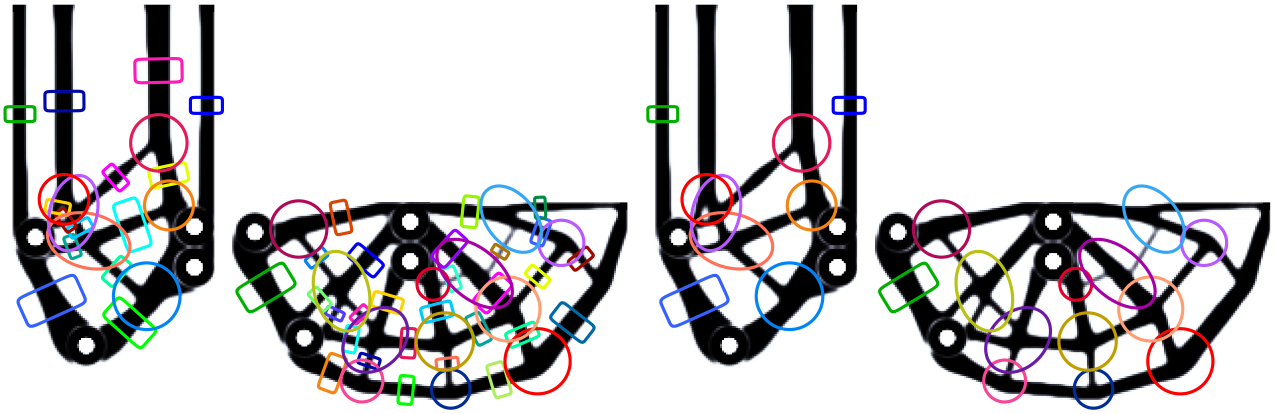
(a) Assembly **(b)** Left part **(c)** Right part

Figure 7.5 – Results for the redundant "thick" L-bracket from stage I with four bolts, minimum distance constraint, and consideration of joint failure.

Identified load paths Figure 7.6 shows the damage shapes obtained by the automatic load path identification procedure. For the full set, 60 damage shapes are present, covering the load paths of the design (14 beams and 6 knots for the left part, 28 beams and 12 knots for the right part). In the reduced set, only knots are kept and in total four straight beams, that are not interrupted by a knot, such that 22 damage cases for potentially critical failures of load paths exist.

Shape optimized design The shape optimized "thick" design is shown in figures 7.7a – 7.7c. This design is constrained by part-wise volume constraints only, hindering the parts to have more volume than they have in the preliminary design from stage I.

For the SO of stage II, failure of one of the four joints or one of the 22 critical load paths is considered. The numerical values for the nominal and worst-case compliances are given in table 7.2. Again, the thick design from stage II improves in all considered aspects and especially in the worst-case compliance for the failure of the most critical load-bearing members, which is cut down to 37%.



(a) Left part, full set **(b)** Right part, full set **(c)** Left part, reduced set **(d)** Right part, reduced set

Figure 7.6 – Identified damage shapes for the redundant L-bracket from stage I involving the full set of shapes (left) or the reduced set (right).

Table 7.2 – Compliance values for the redundant L-bracket example considering damage of joints only or damage of joints and load paths.

Configuration	Figure	$\bar{c}_J^1$	$\bar{c}_{LP}$	c_0	ω_p
4 bolts, stage I, thick	7.5a	137.00	479.05	126.47	35.79%, 39.62%
4 bolts, stage II, thick	7.7a	129.22	177.10	111.47	35.78%, 39.62%
4 bolts, stage II, thin	7.7d	194.84	241.76	146.47	29.98%, 30.00%

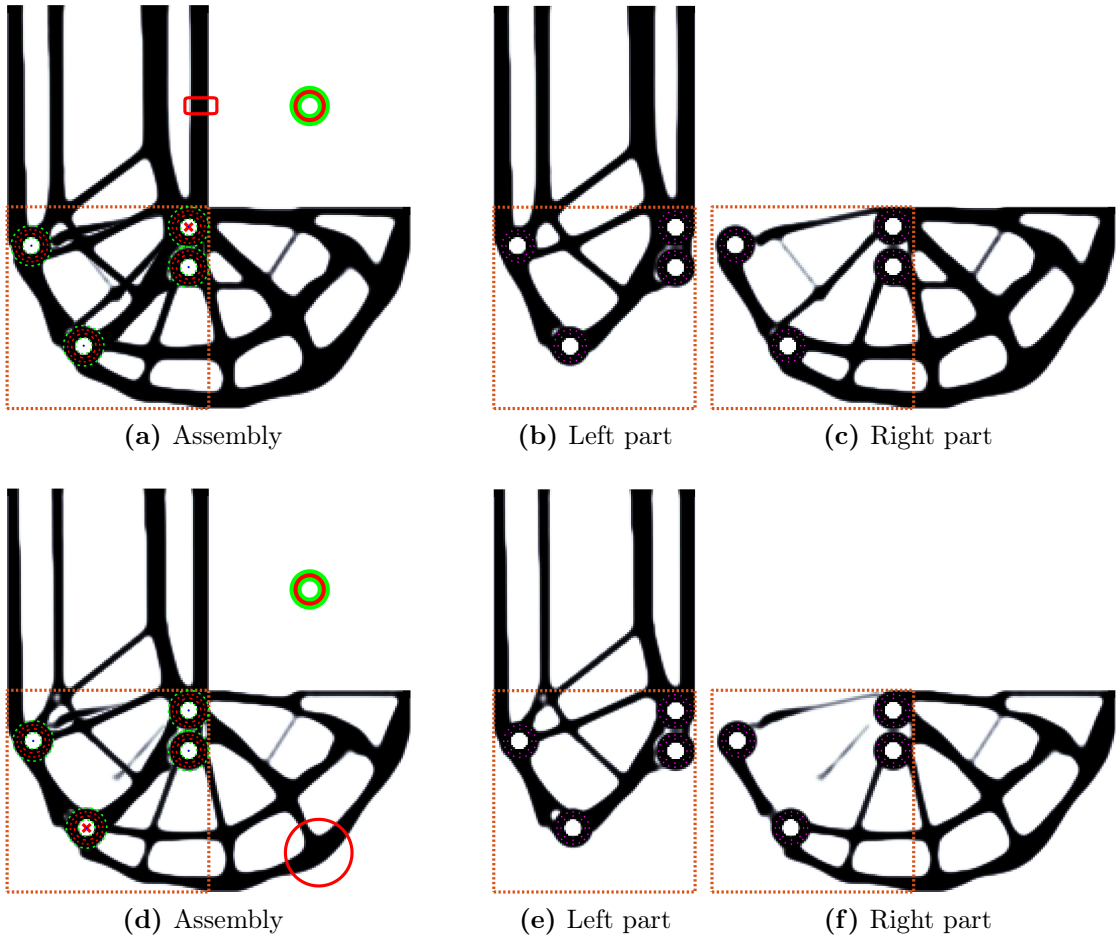
For the preliminary and the shape optimized thick design, the worst-case joint failure is a failure of the joint at the inner corner of the bracket. The worst failure for the structural members however differs between the designs: For the preliminary design, it is located at the right part at the same knot that is marked in figure 7.7d, for the shape optimized thick design it is the rightmost vertical beam of the left part, as shown in figure 7.7a.

The "thin" design, which is constrained to a maximum part-wise volume of 30%, is shown in figures 7.7d – 7.7f. Compared to the thick design, the nominal and the worst-case compliances are higher due to less material available, as listed in table 7.2. However, the assembled design has a way better worst-case compliance than the single-part redundant design from figure 5.13c (cf. tables 5.4 and 7.2). The reason for this is, that the single-part design is susceptible to a failure at the inner corner of the bracket. In contrast to that, for the assembled design using a fail-safe joint layout, this corner is populated by two joints, such that a stronger reinforcement of the corner is achieved. Additionally, only the joints but not their NDS zones were considered as load paths subjected to failure.

7.2.3 Computational effort

Using the two-stage fail-safe optimization procedure presented in section 5.2, stage I involves a constant number of FE analyses to obtain a preliminary design. The computational effort of the subsequent stage II however is then dependent on the preliminary design and grows with the number of identified critical load paths.

For the above examined fail-safe assembly optimization examples, the computational effort for stage I was already higher compared to the single-part fail-safe optimizations treated in



(a) Assembly **(b)** Left part **(c)** Right part

(d) Assembly **(e)** Left part **(f)** Right part

Figure 7.7 – Results for the shape optimized redundant L-bracket after stage II considering joint failure and damage on load paths: "Thick" design (top) and "thin" design (bottom) with less volume. Worst-case damages are marked in red.

section 5.3 since failure of the 4 joints is explicitly considered. Conducting again 200 iterations, in total 800 FE analyses are needed to obtain the preliminary design.

The SO of stage II involves again only 25 iterations due to fast convergence. The number of FE analyses per iterations is set by the number of damage scenarios that are considered, which is in this case the sum of the number of joints and identified critical load paths.

The overall number of FE analyses for the two studied 2D examples are summarized in table 7.3. Compared to an assembly optimization considering joint failure only, the additional computational effort for including the SO stage II, which also includes damage on load paths, is roughly doubled.

Table 7.3 – Total number of FE analyses for the proposed fail-safe assembly optimization involving failure of joints and load paths.

	Damage cases		
Example	stage I	stage II	Total analyses
2D cantilever, redundant	4	43 (4+39)	1 875
2D L-bracket, redundant	4	26 (4+22)	1 450

A possible way to cut down the computational cost of stage I is to not consider joint failure explicitly but rather aim for a damage tolerant joint configuration by using a maximum joint force constraint to redistribute forces evenly across all joints, as demonstrated in the examples of sections 6.8.1 and 6.8.2. However, using this approach, the effect depends on the chosen threshold value $\bar{F}_t$ for the maximum force, which needs to be adapted to the specific problem, such that it is not guaranteed, that in all cases a damage tolerant joint layout is obtained.

8 Summary

The two main topics of this work are the fail-safe structural optimization of single parts and assemblies. Motivated by drawbacks of state-of-the-art methods, algorithms are presented that address these topics in a computationally efficient manner while sticking close to established workflows for topology optimization.

At first, the widely used three field approach for density-based topology optimization is presented and it is shown how to tackle the non-linear topology optimization problem by gradient-based optimization algorithms like e.g. the Method of Moving Asymptotes (MMA).

Afterward, aggregation functions are discussed, which are used intensively for the formulation of the worst-case fail-safe objective and for the application of local volume constraints as redundancy enforcing modification to plain topology optimization. A new aggregation scheme is derived that closes the gap for the aggregation of close to uniformly distributed input values, which is a use case for which known aggregation functions are suboptimal.

Considering the first main topic of fail-safe topology optimization, the largest problem of existing methods that explicitly consider damaged configurations is the required computational time. Therefore, successively methods are assessed to cut down the involved computational cost, which leads to a load-path-based damage model.

A sequential procedure for fail-safe topology and shape optimization is presented, where the main contribution is the second optimization stage which consists of a density-based shape optimization that is capable of handling load-path-based damage in a numerically stable way. By application to 2D and 3D examples, it is demonstrated that this density-based shape optimization significantly improves the performance of a structure under partial failure, which originates from a preliminary topology optimization, while using the same amount of material.

Finally, the focus of the work shifts from single-part optimizations towards the second main topic, namely the simultaneous optimization of components and joints in assemblies. Hence, it is first shown that existing methods adapted from pure topology optimization algorithms cause issues when applied to finer resolution FE models. Consequently, a method that augments a topology optimization by a joint layout optimization, involving the joints' locations explicitly as design variables, is presented. Again, 2D and 3D examples show the applicability and potentials of the method. The proposed assembly optimization method is further augmented by fail-safe aspects in considering failure of joints to obtain assemblies connected in a fail-safe manner. Finally, also the above-mentioned method for considering load-path-based damage is incorporated into the presented assembly optimization method to obtain fail-safe assemblies.

8.1 Conclusion

The most important findings and achieved improvements concerning the two main topics of this work are concluded in the following.

8.1.1 Fail-safe optimization of single components

Three main causes for the high computational cost involved in approaching fail-safe topology optimization problems with existing methods are identified in section 4 of this work, originating from a lack of knowledge about the location of the critical damage region of the optimized part:

1. The number of damage scenarios is either directly coupled to the FE mesh discretization, as in [35], or at least to the dimensionality of the model, as e.g. in [36, 38].

2. Empty or not yet fully converged regions are also considered as damage scenarios, which leads to many unnecessary evaluations before the topology of the part has even settled.
3. After the structural features of the part have evolved, several damage scenarios may only affect a single load-bearing member.

To remedy the problem of unfeasible computational time required for fail-safe topology optimization, especially for 3D problems, this work at first assesses different published methods to cut down the computational effort.

As it turns out, simply excluding a fixed subset of possible damage scenarios from the optimization, as proposed in [36], is not a viable option since it carries the risk of skipping critical damage locations. For the studied cantilever beam example from section 4.2.2, it was shown that even with a relatively fine damage grid level of GL3, despite 833 times the computational effort, this approach led to a higher worst-case compliance at higher total volume than an optimization using redundancy enforcing local volume constraints without any damage considerations at all (see the results in tables 4.4 and 4.1).

In contrast to that, the author's "threshold method" [37], as an adaptive strategy of damage scenario preselection, proves to be conservative and at the same time tackles the second above-mentioned issue concerning empty and unconverged design space regions. However, the highest efficiency improvement reported in this work is a reduction of the overall evaluations to 56% of the original value (see table 4.7). Hence, the achievable reduction by applying the threshold method is not high enough.

Extending the idea behind the threshold method, which aims at reducing the computational cost associated with "empty" design space regions, the load-path-based damage model by the author [37, 58] is presented in section 4.3 as a means to reduce the computational cost also for solid regions of the design. This is accomplished by covering all load-bearing features of the part with a minimal amount of adaptively shaped damage zones, each affecting a single load path. Hence, the number of analyses required for the optimization is then coupled to the count of load paths present in the part and not anymore to the discretization of the FE mesh. Therefore, all three above-mentioned issues are dealt with simultaneously by using the presented load-path-based damage model. Additionally, using a load-path-based damage model brings the fail-safe optimization closer to the requirements defined in CS-25 [14].

To counter convergence issues arising when using load-path-based damage in a topology optimization, in section 5 it is proposed to consider damaged configurations in a shape optimization step only. However, since a shape optimization requires a well-defined geometry already to be present, a two-stage procedure is presented. Therein, a cheap topology optimization without damage considerations is performed as the first stage, followed by the newly proposed density-based shape optimization, which is compatible with the three field topology optimization workflow. The shape optimization proves to be numerically stable and minimizes the number of optimization iterations, in which the cost-intensive damage evaluation is performed. However, for a shape optimization, the design freedom is more restricted than for a topology optimization and the final result is dependent on the preliminary design of the first stage.

The presented two-stage procedure for fail-safe optimization from section 5.2 exploits the fact that redundant designs offer a higher damage tolerance, than designs obtained from a plain topology optimization. Therefore, the first stage aims at optimizing the topology towards a design with redundant load paths, which can be achieved cheaply by using the local volume constraints proposed by [44]. This redundant design is then used as the base design for the fail-safe shape optimization in the second stage. Although the selection of the base design is based purely on engineering judgment, the desired degree of redundancy it offers can be checked early, namely before the cost-intensive optimization towards a fail-safe design is even initiated.

Compared to methods employing the simplified rectangular damage model from [35], the computational effort of the proposed sequential fail-safe optimization approach is smaller by many orders of magnitude which makes it feasible for industrial applications. For the 2D and 3D problems considered in section 5.3, the total number of FE analyses required for a fail-safe optimization of designs with redundant load paths is reduced by a factor of 1500 – 2000, depending on the example, as reported in table 5.7. By applying the further reduction of damage scenarios discussed in section 5.4, additional savings of about 60% are achieved, such that the total number of analyses goes even below 1000, as listed in table 5.8. In combination, this corresponds to roughly only 5.6 times (redundant 2D cantilever beam) or 4.3 times (redundant 2D L-bracket) the computational effort required for the presented fail-safe optimization when compared to a plain topology optimization.

In relation to the implicit approach of increasing damage tolerance solely by enforcing redundancy with local volume constraints (see [44]), the fail-safe performance of the sequentially optimized structures improves significantly within a few additional iterations of density-based shape optimization. For the 2D examples from section 5.3, reductions of the worst-case compliance of 50% and 68% at constant volume are achieved within 25 iterations, reductions of 38% and 50% are reported for the 3D examples considered. Counting in also the results from section 7.2, even the nominal compliance for the undamaged case has improved in three out of four redundant 2D designs studied.

In summary, the results of this work underline that the sequential approach of redundant topology optimization with local volume constraints and the subsequent density-based shape optimization is an efficient procedure to design fail-safe structures. The initially set goal of performing fail-safe optimization with a moderately increased computational effort by keeping the total number of FE analyses at most one order of magnitude higher than for a plain topology optimization is achieved.

8.1.2 Fail-safe optimization of assemblies

In contrast to single-part topology optimization, the optimization of assemblies gained only relatively little attention in the literature. Existing methods for assembly optimization suffer from at least one of the following main issues concerning the modeling of the joints:

1. Fastener positions are fixed throughout the optimization.
2. Fasteners are defined as groups with a fixed layout.
3. Fasteners can only be placed at discrete, predefined positions along a fixed grid.
4. Fasteners are not discrete and/or clustered.
5. Different types of connections cannot be distinguished.

In this work, a new method for the simultaneous optimization of components and their connections inside assemblies is presented in section 6. A short discussion in section 6.1.1 reveals drawbacks of assembly optimization approaches relying solely on topology optimization methods, like e.g. the method from [47]. Motivated by this, an extension is proposed in section 6.1.2 in which joints are included in the numerical model as discrete features with their own continuous position variables that are considered in the set of design variables to be optimized. Simultaneously, the material distribution of each component is optimized according to the three field topology optimization approach. Additionally, void or solid movable non-design spaces may be defined for each joint, thus coupling the local geometry of the components to the joints' positions.

The presented approach for assembly optimization allows specifying the exact amount of joints as well as the local joint geometry, the stiffness properties, and the force transfer zone

area for every joint. Since the joints are also continuously movable, all above-mentioned issues are tackled at once with the proposed method.

Examples of the application are given in section 6.8, where e.g. assembled two-part models for the 2D cantilever beam and L-bracket problems are presented, showing only a small penalty of 6% to 9% in terms of compliance compared to the corresponding single-part reference designs from a plain topology optimization (see tables 6.1 and 6.4).

Furthermore, for all two-joint 2D examples presented, the joints are placed on the main thick load paths of the parts, underlining their function as interconnecting features vital for the load transfer between the parts of the assembly. Though the number of joints present in the model can be arbitrarily increased, it is not guaranteed, that any "excess" joints are connected to the individual parts in a meaningful way or transfer load at all, as shown by some four-joint 2D examples. Hence, a design with additional joints may be as susceptible to a single joint failure as a design with only a minimal amount of joints.

To counter this, methods are presented to increase the robustness of optimized assemblies towards the failure of single or even multiple joints. At first, an implicit way is presented that relies on constraining the maximum force in each joint to a deliberately low value, such that forces need to be distributed more evenly across all joints. This enforces all joints to be involved in the load transfer. Additionally, also an explicit way is presented, considering failure of joints directly as failure scenarios during the optimization (see section 6.7), which yields assemblies connected in a fail-safe manner at a moderately increased computational cost since the number of joints is usually relatively low.

In any case, the number of joints to be used in the presented method has to be predefined. On the one hand, this could be a disadvantage, since choosing a too high number of joints may lead to results with unconnected or inefficient joints, as some results for a plain assembly optimization show. Still, on the other hand, this can also be seen as an advantage, as it offers the user full control over the exact number of joints in the model, a feature not inherent to all methods of assembly optimization in the literature. Nevertheless, as just discussed, there exists always the solution to put inefficient joints to good use in increasing the robustness of the assembly concerning joint failure.

Since assemblies that have a damage tolerant joint pattern are still susceptible to damage on critical load paths of the individual components, the method developed in section 5 for an efficient fail-safe optimization of single parts is incorporated into the presented assembly optimization approach, which is discussed in section 7.1. The obtained results for fail-safe assemblies with redundant load paths and redundant joint patterns are finally shown in section 7.2. The numbers given in table 7.3 reveal, that for the shown examples the computational effort for additionally considering damage on load paths is roughly doubled compared to an assembly optimization considering damage of joints only. Since the total number of analyses stays in the lower thousands, it is expected, that also large-scale models are solvable within reasonable computational time.

Though the method for fail-safe assembly optimization is only demonstrated for connections of two parts in this work, it can be applied to assemblies consisting of an arbitrary number of components without modifications. These components can either share a single connection region or multiple distinct connection regions may be present.

A general major advantage of optimizing assemblies as a whole is, that by optimizing the joint locations also the distribution of the interface loads acting between the parts is optimized in the final result. By considering material and clearance zones for joint mountability already in the numerical model, a consistency between the simulated and manufactured part geometry is obtained, which is another advantage. Additionally considering the failure of joints and load paths allows for trading off nominal performance in favor of residual performance under failed

conditions, which may significantly be improved at a moderately increased computational cost.

8.2 Outlook

The application of the proposed two-stage procedure for fail-safe optimization from section 5 requires a certain level of engineering judgment: In stage I of the optimization, a suitable preliminary design has to be found via topology optimization methods. Although the presented results underline, that the performance under partial failure of every design is improved in the newly proposed density-based shape optimization of stage II, the greatest benefit is obtained when the preliminary design already provides good robustness to partial failures. In this work, it is recommended to choose a redundant design, however, the degree of redundancy is evaluated by visual inspection only, a systematic way to quantify it still needs to be developed. Different parameters for the involved local volume constraints result in different topologies with a varying number of distinct load paths. If in doubt, rather a design with too many redundant load paths should be chosen. Besides leading to a higher computational effort for the subsequent stage II, this should not have any further significant negative side-effects in contrast to the choice of a design with too few and thus not fully redundant load paths. Nevertheless, even existing methods for fail-safe optimization such as the method of Jansen et al. [35] require the "correct" choice of parameters, otherwise, only non-redundant designs may be obtained. For the presented two-stage procedure, a first inspection of the preliminary design can at least be made at an early stage.

Fail-safe optimizations in 3D are in this work only shown for lattice-type designs. The decomposition of these designs into load paths is straightforward and does not differ from 2D problems. How to treat shell-like 3D structures and define load paths therein however remains an open question, since these designs may be composed of one shell only (consider e.g. a tube as optimal design for a torsional load). It is known, that lattice-type 3D designs have suboptimial stiffness properties for the intact case [2]. Nevertheless, lattice designs offer some advantages compared to shells from a practical point of view in terms of damage tolerance, e.g. when considering crack propagation that could cause partial failure in the first place. Similar to the 2D case, by increasing the "porosity" of a design, an increased robustness with respect to partial failure is also expected for 3D cases. Porous lattice-type structures could be interpreted as the equivalent of open-cell microstructures. While closed-cell microstructures have been shown to have optimal stiffness (see [2]), at least for the intact case, the manufacturability of equivalent structures on a macro-scale (e.g. up-scaled versions of metal foams) is anyway questionable.

Fail-safety is in this work only addressed in terms of compliance. Primarily, the optimization formulation of Jansen et al. [35] is followed, in which the worst-case compliance is minimized, while the volume of the part is constrained to a maximum value. In section 5.5, the developed methods are also applied for optimizations involving volume minimization under a maximum compliance constraint. However, structural failure is often evaluated in terms of exceeding allowed stresses and not displacements. Nevertheless, stress-minimization or even stress-constrained topology optimization is a challenging topic in itself, even without considering damage scenarios, since stress is a local measure. The focus of this work was primarily on developing a tool to cut down the computational cost for optimizations including partial failure, as a prerequisite before being able to move over to more challenging objective functions. However, with the presented two stage-procedure, at least stress-minimization optimizations considering failure of load paths should be realizable.

The shape optimization stage of the presented fail-safe optimization procedure is performed via a density-based method to ensure maximum compatibility with existing topology optimization

codes. Additionally, the required implementation effort is minimized. As with density-based topology optimization, the obtained designs however still require a geometrical interpretation or reconstruction in order to be CAD compatible. An alternative would be to perform the reconstruction already after stage I and perform the fail-safe shape optimization of stage II directly on a CAD-compatible isogeometric FE model. However, isogeometric analysis poses problems of its own when it comes to volumetric meshing.

For the presented assembly optimization method, this work focuses on "point-type" connections, like spot welds, rivets, or bolts, for which the location is described by a single reference point per joint. To extend the applicability of the proposed method, also other types of connections could be implemented. The shown method is, with some extensions, also applicable to "line-type" connections like e.g. weld seams or glue lines. An open polygon or an open spline curve would be the reference curve for these types of connections. The position design variables are in this case the coordinates of the polygon's edges or the spline's control points. Again, a pattern of springs is attached to the reference curve to model the size of the force transfer zone. A dilated version of the reference curve can be used to incorporate the minimum material zone, which would again be projected into the pseudo-density field of the topology optimization.

Considering the achieved computational savings by the presented algorithms for fail-safe design, an application of the developed methods is also possible for high-resolution large-scale FE models, with a complexity way beyond the shown relatively simple academic examples. Due to the compatibility with existing topology optimization approaches, also other aspects from robust topology optimization (RTO) can be included in the optimization, such as e.g. the consideration of a scatter of the applied loads or the modeling of other uncertainties.

A Additional theory on aggregation functions

This appendix provides additional information and analytically derived relationships to the aggregation strategies discussed in section 3 of this work. Also, a numerical study of the newly derived aggregation function A_{uni} for uniformly distributed input values is given.

A.1 Influence of the aggregation parameter p on combined gradients

The aggregation functions presented in section 3 are approximations to the max-operator. For increasingly larger aggregation parameters p, the aggregated output value is dominated by the input term with the maximum value and all other input terms with lower values quickly lose on influence on the aggregated value.

Even more important for gradient-based optimization is how the combined gradient of the aggregated output is assembled from the individual gradients of the input terms, which is analyzed in the following short study.

As a test set, an input vector $\boldsymbol{\tau}$ with only two entries to be aggregated is studied. The first value is considered to be the maximum value $\bar{\tau} > 0$, while the other value only reaches a fraction of this maximum value. Introducing a scaling factor $s \in [0, 1]$, the test set thus is:

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} \bar{\tau} \\ s\bar{\tau} \end{bmatrix} \tag{A.1}$$

By using an aggregation function A, the aggregated term τ_{agg} is obtained:

$$\tau_{agg} = A(\boldsymbol{\tau}) \tag{A.2}$$

As shown in equation (3.2), the gradient of τ_{agg} with respect to a DV is a weighted linear combination of the original individual gradients of the input terms. In this case, a factor w_1 is obtained as the weight for the gradient of τ_1 and a corresponding second factor w_2 as the weight for the gradient of τ_2 via the equations given in Appendix D.6:

$$\frac{d\tau_{agg}}{d\varrho_i} = \frac{dA(\boldsymbol{\tau})}{d\varrho_i} = w_1 \frac{\partial \tau_1}{\partial \varrho_i} + w_2 \frac{\partial \tau_2}{\partial \varrho_i} \tag{A.3}$$

The relative difference in between the two input values τ_1 and τ_2 from equation (A.1) can now be varied by altering the scaling factor s and the resulting influence on the gradient of τ_{agg} can then be analyzed taking into account the proportion of the weights w_2/w_1.

Since for the p-norm and p-mean aggregation function the scale of the input values can be pulled out (in this case $\bar{\tau}$), the result is only dependent on s and the aggregation parameter p:

$$\left(\frac{w_2}{w_1}\right)_{pn/pm} = s^{p-1} \tag{A.4}$$

The same relation also holds for the other "p-like" aggregation functions introduced in this work, like e.g. A_{uni} from section 3.4.3.

For the KS aggregation function however, the result is also dependent on the absolute scale of the values:

$$\left(\frac{w_2}{w_1}\right)_{KS} = \mathrm{e}^{p(s-1)\bar{\tau}} \tag{A.5}$$

The relative decay of the influence weights for the p-norm and p-mean aggregation functions dependent on the scaling factor s is shown in figure A.1 for various exemplary p values. Nevertheless, for the curves in figure A.1 referring to the KS function, the whole maximal argument $\hat{p} = p\bar{\tau}$ is altered since the KS function is scale-dependent.

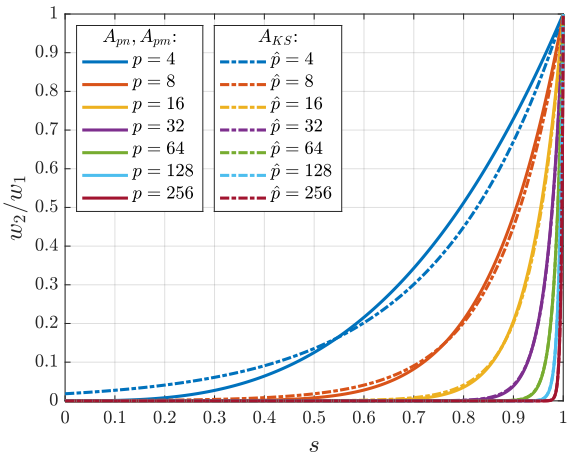


Figure A.1 – Decay of the influence weight ratios for the p-norm, p-mean, and KS aggregation functions for different aggregation parameters p and $\hat{p}$.

Figure A.1 highlights, that already from medium values for p or $\hat{p}$ on, the influence of the gradient belonging to the second largest aggregated value (described by w_2) becomes quickly negligible when compared to the weight of the dominating, largest value (described by w_1).

Table A.1 lists the necessary minimal aggregation function parameters p or $\hat{p}$ to achieve a desired drop of the influence factor ratio by one order of magnitude (meaning $w_2/w_1 = 0.1$) for some exemplary s values. Note, that these values are also valid for input sets $\boldsymbol{\tau}$ with more than two input terms.

Table A.1 – Minimal aggregation parameters p or $\hat{p}$ to obtain a drop of influence factors by one order of magnitude for certain input value fractions.

Drop by one order for $s = \ldots$	Minimal p for A_{pn}, A_{pm}	Minimal $\hat{p}$ for A_{KS}
0.9	22.85	23.03
0.95	45.89	46.05
0.99	230.11	230.26

The shape of the curves from figure A.1 and the values in table A.1 underline, that the aggregation characteristics regarding the combination of gradients of p-like functions and the KS function are very similar. The major difference is, that the slope of the curves for the KS function at $s = 1$, which neighborhood is usually the region of interest, is with a value of $\hat{p}$ slightly higher

than the slope of $p-1$ for p-like functions. Another aspect usually not relevant in practice is, that with the KS function all terms have a (possibly very small) non-zero contribution to the combined gradient, even at $s=0$, whereas for p-like functions the weighting factor in this case becomes zero.

A.2 Data fitted derivation of correction factors for p-like aggregation functions

The general aggregation function A_g from equation (3.23) offers an additional parameter α that allows deriving matching aggregation functions for input vectors $\boldsymbol{\tau}$ following different value distributions.

This study will focus on input vectors with non-negative values that are bounded in an interval $[0, \bar{\tau}]$. The goal of such a matching aggregation function is to retrieve the maximum value $\bar{\tau}$ of the input vector, ideally with zero error. At first, only uniformly distributed input values are considered.

Similar to the procedure applied in section 3.4.1 to find ideal input values, now the ideal value of α is sought to get zero approximation error. Consequently, the term inside the large bracket of equation (3.23) has to reach a value equal to $\bar{\tau}^p$:

$$\alpha \underbrace{\sum_{i=1}^{n_t} \tau_i^p}_{S} \stackrel{!}{=} \bar{\tau}^p \tag{A.6}$$

The sum term in the above equation is referred to as S in the following derivation. To gain a better understanding of p-norm based aggregation functions, let g_d be a discrete function dependent on an integer index variable i, taking the input values τ_i to the power of p:

$$g_d(i) = \tau_i^p, \quad i = 1, \ldots, n_t \tag{A.7}$$

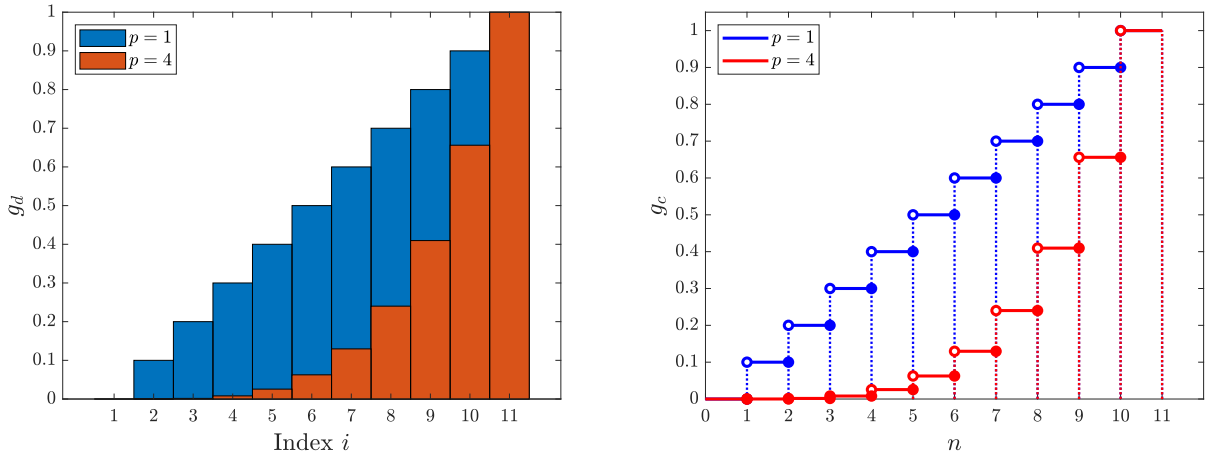
Figure A.2a shows a graphical representation of g_d if the input values τ_i are 11 equally distributed numbers between 0 and 1 (with a step-size of 0.1) and for exemplary p values of 1 and 4. For the following derivation, without loss of generality, the values τ_i are sorted in ascending order and the maximum value $\bar{\tau}$ is 1, since any scaling factor can be pulled out (see equations (3.15)).

The term S of equation (A.6) can now also be written as:

$$S = \sum_{i=1}^{n_t} \tau_i^p = \sum_{i=1}^{n_t} g_d(i) \tag{A.8}$$

In the case of $p=1$, S simply yields the sum of all input values τ_i. For higher p, the values τ_i are scaled progressively prior to summing. The value of S corresponds to the area of the bars shown in figure A.2a when a bar width of 1 unit is assumed[23].

[23] An inverse mapping with the p-th root of S yields the p-norm of the input values. Since for larger p the total area of the bars is dominated by the area of the largest bar, the accuracy of the p-norm aggregation function consequently increases in this case.



(a) Discrete function g_d **(b)** Piece-wise defined function g_c

Figure A.2 – Function values of g_d and g_c for equally spaced input values with $p = 1$ and $p = 4$ for an example with $\bar{\tau} = 1$.

The discrete function g_d can also be rewritten as a piece-wise defined function g_c with a continuous "index" parameter n:

$$g_c(0) = \tau_1^p = \text{const} \tag{A.9a}$$
$$g_c(n) = \tau_i^p = \text{const}, \quad \text{for} \quad i - 1 < n \leq i, i = 1, \dots n_t \tag{A.9b}$$

The function g_c is shown in figure A.2b. The area between g_c and the "index" n axis can be obtained by integration. Since the "bars" or in this case steps are again 1 unit wide, the same value as S from equation (A.8) is obtained:

$$S = \int_0^{n_t} g_c(n) dn \tag{A.10}$$

For the following calculations, the piece-wise defined function g_c is approximated by a smoothed function g_s. The function g_s is constructed by taking a smooth modeling function $\tau(n)$ to the power of p:

$$g_s(n) = \tau(n)^p, \qquad n \in [0, n_t] \tag{A.11}$$

The function τ has to be chosen such, that it models a specific distribution of expected input numbers. To model uniformly distributed numbers in an interval of $[0, \bar{\tau}]$, τ is simply a linear function. Without the loss of generality, τ is again chosen to be monotonically increasing with the "index" n:

$$\tau(n) = \bar{\tau} \frac{n}{n_t}, \quad n \in [0, n_t] \tag{A.12}$$

Examples for the resulting smooth functions g_s for $p = 1$ and $p = 4$ are shown in figure A.3.

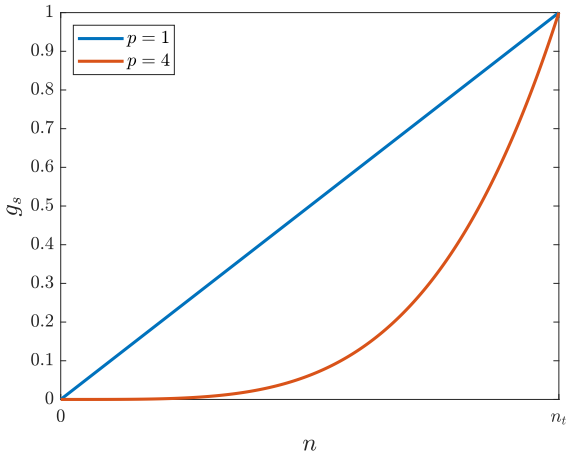


Figure A.3 – Smoothed functions g_s with $p = 1$ and $p = 4$ for an example with $\bar{\tau} = 1$.

With an increasing number n_t of input values, the integral of the piece-wise defined function g_c in the interval $[0, n_t]$ approaches the corresponding integral of the smooth function g_s. This can be best seen if performing a linear coordinate transformation, that compresses the n-values from the interval $[0, n_t]$ to fit into a normalized range of $\tilde{n} \in [0, 1]$. The mapping function is then:

$$n = n_t \tilde{n}, \quad dn = n_t d\tilde{n} \tag{A.13}$$

The integral over g_c can thus also be evaluated on the normalized index coordinates $\tilde{n}$:

$$S = \int_0^{n_t} g_c(n) dn = n_t \int_0^1 g_c(n_t \tilde{n}) d\tilde{n} \tag{A.14}$$

The width of each step of the function g_c in $\tilde{n}$ coordinates becomes smaller, the higher the number n_t of input terms is since all $\tilde{n}$-values must fit in the range of $[0, 1]$. In the limit of $n_t \to \infty$, the steps are infinitesimally thin, such that they are not distinguishable from the smooth function g_s.

Consequently, also the difference between the integral over g_c and over the smooth approximation g_s is getting negligible. For large n_t, the value of S from equation (A.14) can thus be approximated by:

$$S \overset{n_t \to \infty}{=} \int_0^{n_t} g_s(n) dn = \int_0^{n_t} \tau(n)^p dn = n_t \int_0^1 \tau(n_t \tilde{n})^p d\tilde{n} \tag{A.15}$$

Inserting the modeling function τ for uniformly distributed values form equation (A.12) yields an expression for S:

$$S \overset{n_t \to \infty}{=} n_t \int_0^1 \tau(n_t \tilde{n})^p d\tilde{n} = n_t \bar{\tau}^p \int_0^1 \tilde{n}^p d\tilde{n} = \bar{\tau}^p \frac{n_t}{p+1} \tag{A.16}$$

By using equation (A.6), the unknown value for the correction factor α for uniformly distributed input values can now be determined:

$$\alpha = \frac{\bar{\tau}^p}{S} = \frac{p+1}{n_t} \tag{A.17}$$

The specific aggregation function proposed in this work for uniformly distributed values is therefore:

$$A_{uni}(\boldsymbol{\tau}) = \left(\frac{p+1}{n_t} \sum_{i=1}^{n_t} \tau_i^p \right)^{\frac{1}{p}} \tag{A.18}$$

Note that $\bar{\tau}$ in the above equations (A.12) – (A.17) is the unknown non-negative maximum value of the input values, that is to be retrieved by the proposed aggregation function. Since $\bar{\tau}$ canceled out in equation (A.17), no prior knowledge of $\bar{\tau}$ is needed. However, it will still be retrieved by using A_{uni}, as long as the assumption of uniformly distributed input values is met.

The modeling function τ in the above derivation was chosen to yield equally distributed values in the desired range of $[0, \bar{\tau}]$ if it is evaluated at equidistant n-values. This sampling can also be performed on the normalized $\tilde{n}$-values, in which case the procedure equals the inverse transform sampling technique [115] and the modeling function τ in $\tilde{n}$-coordinates is seen as the inverse CDF of the desired distribution.

To be able to apply the above procedure to derive correction factors α also for moderately skew deviations from a uniform distribution, an exponent q is introduced to the modeling function τ from equation (A.12):

$$\tau(n) = \bar{\tau} \left(\frac{n}{n_t} \right)^q, \quad n \in [0, n_t] \tag{A.19}$$

Figure A.4 shows the inverse CDFs (which is $\tau(n_t \tilde{n}) = \bar{\tau} \tilde{n}^q$), CDF and PDF functions of the corresponding modeled input distributions for exemplary q exponents.

Performing the calculation steps from equations (A.16) and (A.17) with the modeling function τ from equation (A.19) yields the correction factors α for the corresponding matching aggregation functions for the shown skew distributions:

$$\alpha = \frac{\bar{\tau}^p}{S} = \frac{pq+1}{n_t} \tag{A.20}$$

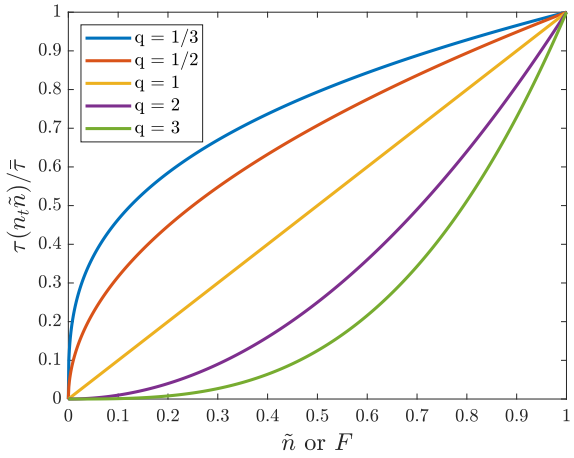
Together with the well-known p-norm (A_{pn}) and p-mean (A_{pm}) aggregation functions, which are ideal for the most extreme skew input distributions, the derived p-like aggregation functions with correction factors α as given in equation (A.20) allow for a broader application of aggregation, offering high accuracy also for uniform or moderately skew input distributions.

A.3 Numerical test of the A_{uni} aggregation function

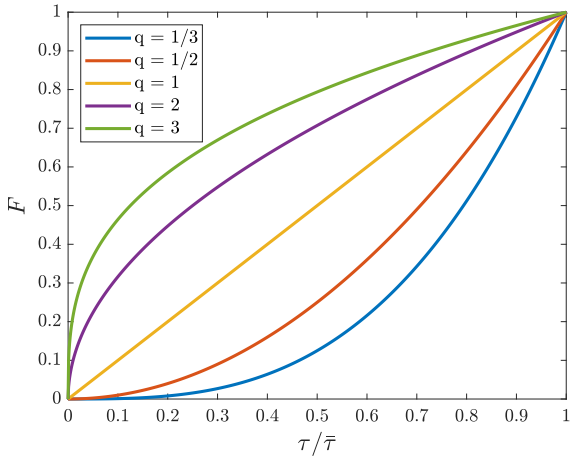
For the numerical test of the aggregation function A_{uni} from section 3.4.3, vectors $\boldsymbol{\tau}$ with n_t random, uniformly distributed numbers in the range of $[0, 1]$ are constructed[24].

In this parameter study, the count of sampled numbers n_t is varied in decades from 10 to 10^6 and the approximative property of A_{uni} for increasingly larger aggregation parameter values p is examined and compared to the ones from the known p-norm (A_{pn}) and p-mean (A_{pm}) aggregation functions.

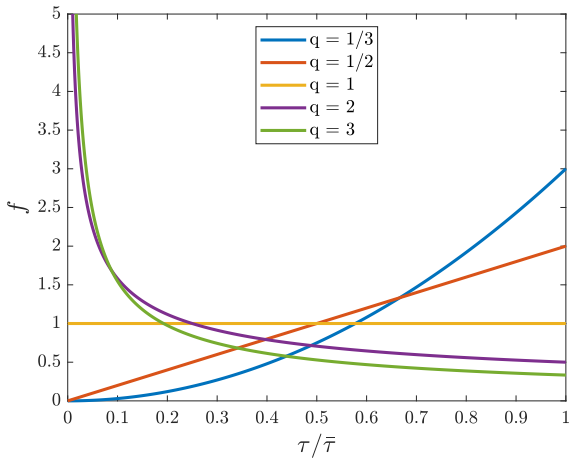
[24]Since any positive scaling factor can be pulled out of A_{pn}, A_{pm}, and consequently also A_{uni} (cf. section 3.3.2), all following findings are also valid for scaled input intervals.



(a) Normalized modeling functions or inverse CDFs



(b) CDFs



(c) PDFs

Figure A.4 – Inverse CDFs, CDFs, and PDFs of normalized modeling functions $\tau/\bar{\tau}$ for different q values.

Since all three aggregation functions converge to the real maximum of the input values with increasing p, the largest differences are expected for low p values. Therefore, a broad range of p values, starting with a low value of $p = 1$ as a worst case and doubled up to a final value of $p = 1024$, are considered.

The p-norm aggregation function A_{pn} is known to overestimate the real maximum. For the case of $p = 1$, A_{pn} simply yields the sum of input values. Since the expected value of the input distribution is 0.5, the aggregated value tends for a large count n_t towards:

$$A_{pn}^{p=1}(\boldsymbol{\tau}) \to \frac{n_t}{2} \tag{A.21}$$

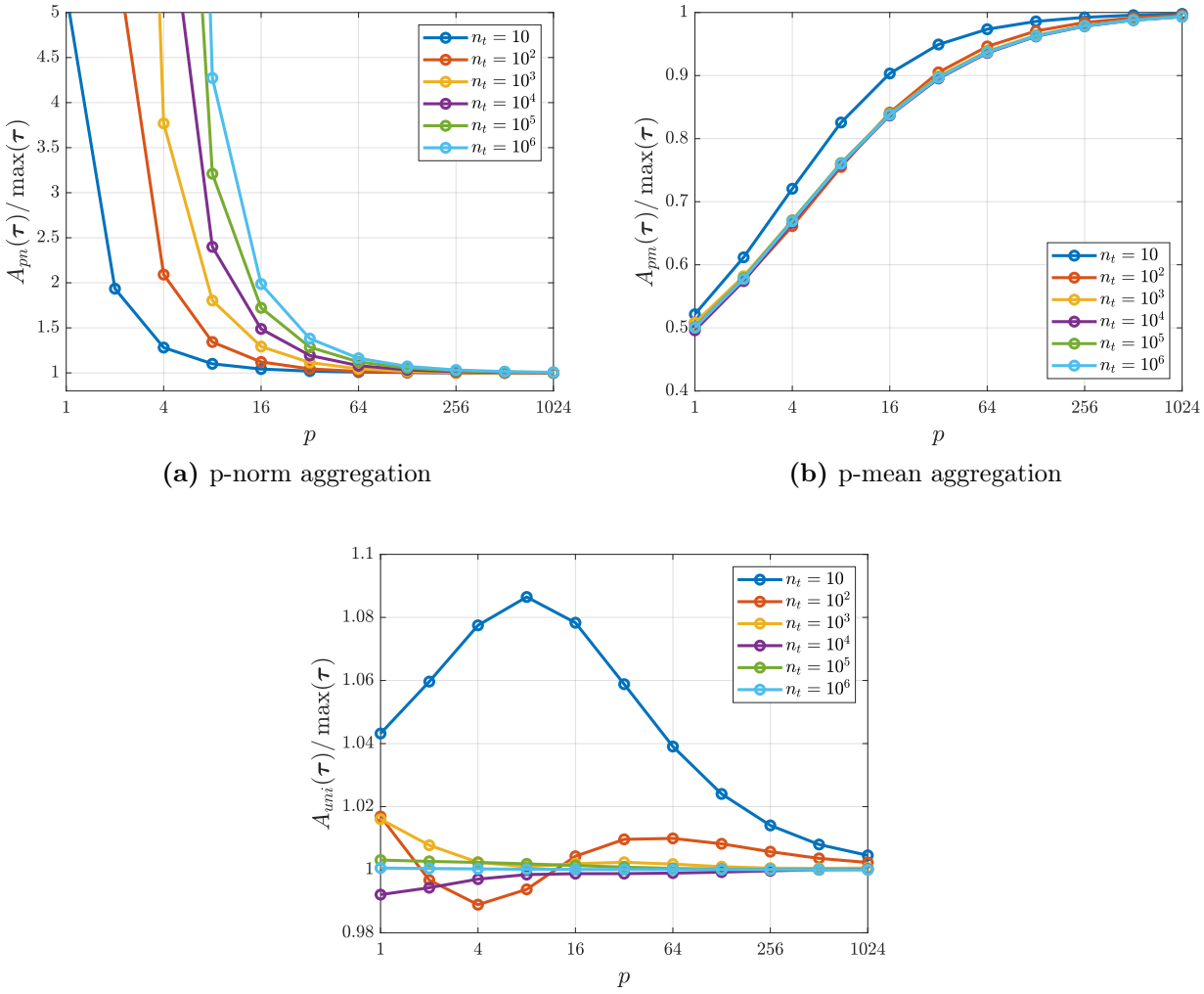
Therefore, A_{pn} severely overestimates a uniform distribution by a factor of $n_t/2$ when using the given low p value. Although the error decreases with larger p as shown in figure A.5a, the performance again diminishes for a larger count of input values n_t. For the studied cases, a value of at least $p = 16$ is needed to stay below a relative error of 100% when using A_{pn}.

The p-mean aggregation function A_{pm} on the other hand underestimates the real maximum

value and compensates for the number of input values, as long as they follow the same distribution. For $p = 1$, A_{pm} yields the arithmetic mean that tends towards the expected value of the uniform distribution for large n_t:

$$A_{pm}^{p=1}(\boldsymbol{\tau}) \to \frac{1}{2} \tag{A.22}$$

As shown in figure A.5b, a relative error of 50% is obtained for the low p value range and the performance of A_{pm} is not penalized by larger n_t counts. Still, a value of $p = 32$ is just not enough to reduce the relative error of A_{pm} below 10% for the studied cases.



(a) p-norm aggregation

(b) p-mean aggregation

(c) p-uniform aggregation

Figure A.5 – Comparison of the aggregation of randomly picked uniformly distributed input values for different sample sizes and aggregation parameters p.

The A_{uni} aggregation function is specifically derived for uniformly distributed input values and therefore clearly outperforms the p-norm and p-mean aggregation functions in this study. As shown in figure A.5c, the accuracy of A_{uni} increases, the closer the random input samples approach a perfectly uniform distribution, which is here the case for larger n_t. Then, independently of the choice of p, a low error of under 2% (for $n_t \geq 100$) is observed. Nevertheless, it cannot be

generally said that A_{uni} is over- or underestimating for any choice of input values, only that it also converges to the real maximum for large p.

The results from figure A.5 emphasize that A_{pn} is virtually unusable for low p values and high n_t counts. Also, A_{pm} may require a very large p in the order of hundreds to achieve sufficient accuracy for input values that span a wide interval, like it is e.g. the case for the studied uniformly distributed values. The overall lowest errors at low to moderately high p values are observed in this case for the newly proposed aggregation function A_{uni}.

B Additional theory and studies on load-path-based fail-safe optimizations

This appendix discusses details related to the theory and application of the load-path-based damage model presented in section 4, which is used throughout this work to model partial failure in the context of density-based fail-safe SO introduced in section 5.

B.1 Automatic identification of load-bearing members

The presented methods for fail-safe optimization from sections 4, 5, and 7 use a set of parametric geometrical shapes from which damage scenarios are derived. Ideally, these shapes can be extracted from an existing design by automated algorithms. Therefore, this work proposes to subdivide lattice-type results of a TO into its load paths, consisting of "beams" and "knots".

Identifying trusses from topology optimized lattice-type designs has been studied before, for instance as a design tool to support the manual CAD interpretation of the obtained design [116], as a tool to derive a manufacturable design by taking the TO result as inspiration [117], or as a means to incorporate explicit feature size control during the TO [118]. However, as already mentioned, the motivation for the member identification in this work is different and also "knot" regions of the design are of interest.

Two identification methods successfully applied by the author are briefly presented: The stress criterion method [37] and a method based on image processing algorithms [58]. These methods may be substituted by any other algorithm offering the same functionality or even a manual subdivision can be performed.

The stress criterion method is discussed in Appendix B.1.1. This method reuses the results of a FE analysis to classify all finite elements of the solid phase into the two categories of "beams" and "knots". Then, the elements are grouped in clusters and auxiliary member shapes are calculated. The total set of auxiliary shapes approximates the original part in a skeleton-like way.

In contrast to that, the method based on image recognition discussed in Appendix B.1.2 goes the opposite way. It does not rely on FE results but instead on pure pixel data (or voxel data in 3D) of an image of the part's geometry. From the image, a skeleton is derived, which is then divided into its parts. Finally, the original finite elements are mapped to their respective closest skeleton part and the auxiliary member shapes are calculated.

The application of the two methods is demonstrated on the simple cantilever design from figure C.4a found in Appendix C.4, which resembles the result used in the original paper by the author [58]. More complicated topologies are dealt with in [58], and in sections 5.3 and 7.2.

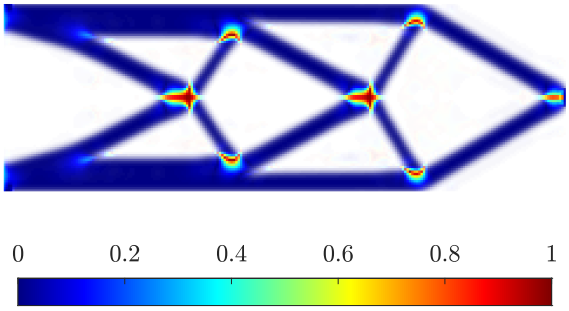
Appendices B.1.1 and B.1.2 describe the details of the two aforementioned methods for an automatic load path identification. Appendix B.1.3 covers the parameter extraction for the identified load-bearing members, such that suitable auxiliary member shapes can be constructed, as described in section 4.3.4.

B.1.1 Stress criterion method

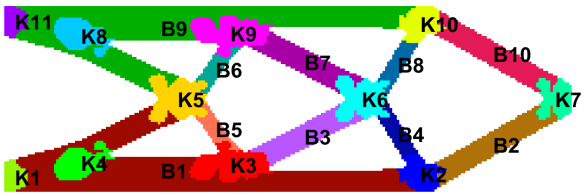
The stress criterion method is a heuristic subdivision approach, exploiting the observation, that for compliance minimization TO problems with a single load case material tends to accumulate along the major principal stress direction (cf. [44]). Hence, a measure for the uniaxiality of the stresses is employed as a classification criterion. Since the stresses inside a structure are mostly anyway of interest, no additional FE simulation is needed and no special algorithms have to be used. Also, irregular meshes can be processed without additional effort.

The basic load path identification algorithm consists of the following steps:

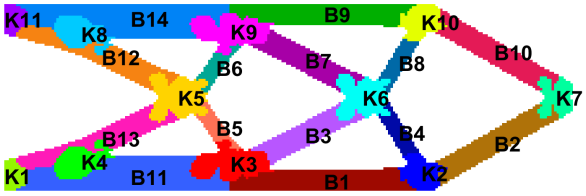
1. Classification of elements into classes “beam” and “knot”
2. Clustering of adjacent elements of the same class
3. Post-processing of the clusters



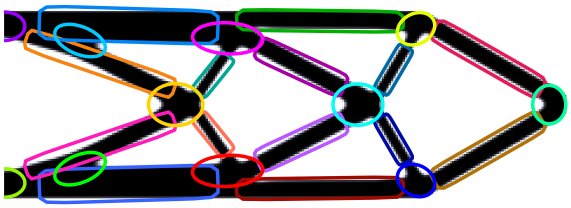
(a) Normalized stress criterion values φ_n of solid phase



(b) Identified clusters: Beams “B” and knots “K” before splitting



(c) Clusters after splitting



(d) Resulting auxiliary member shapes on density field

Figure B.1 – Load path identification with the stress criterion for the cantilever beam example.

Classification: It is assumed that the structural beams are in a uniaxial stress state. For each finite element, the criterion uses the stress state $\boldsymbol{\sigma}$ evaluated at the element’s center point. The normalized criterion value is calculated as:

$$\varphi_n(\boldsymbol{\sigma}) = \frac{4}{\pi} \arctan(\underline{\sigma}/\overline{\sigma}) \tag{B.1}$$

Here, $\overline{\sigma}$ is the highest principal stress magnitude (in absolute values) and $\underline{\sigma}$ is the second-highest principal stress. To limit the range of values of the criterion, a scaled arctan function is used to map the result to the interval of $[0, 1]$. Low values of φ_n indicate a strongly directional stress state, whereas in the extreme case the value of φ_n becomes 1, if the principal stresses

are of equal magnitude, indicating a mixed stress state. Figure B.1a shows the stress criterion values for a 2D cantilever beam example.

The classification into beams and knots is then done by choosing a threshold value for the criterion, for example based on the empirical cumulative distribution function of the criterion values. For figure B.1b, the threshold was chosen such that 70% of the elements were classified as belonging to beams and the remaining 30% to knots.

Clustering: Neighboring elements that are classified as the same type (beam or knot) are grouped into clusters. This is done by using a recursive algorithm iterating over the direct neighbors of an element until all solid elements are processed once.

Post-processing: In a post-processing step, very small clusters under a desired minimal value are merged with the largest surrounding cluster. The result from figure B.1b is obtained. Undesirably large beam clusters "B1" and "B9" can be seen in the top and bottom regions. These overly large clusters may occur at locations, where a knot cluster intersects a beam along its length but does not fully cut through it. The beam cluster needs then to be split at this position. An example of that is knot "K3", where the bottom beam in the final post-processed result in figure B.1c is split into two separate beams "B1" and "B11" (the same accounts for the top with "K9", "B9" and "B14").

Branched beams are not desired either. They can be easily identified since for them the dominant principal stress direction is not constant within the cluster. This information is used to subdivide the cluster by choosing a threshold value on the inclination angle by Otsu's method [119]. In this example this was the case for the beams "B1" and "B9" from figure B.1b which are branched near the left side. In the final result in figure B.1c, "B12" and "B13" resulted from this splitting procedure.

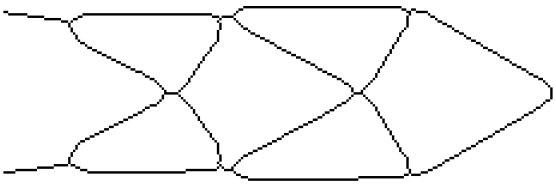
Finally, by applying the parameter extraction discussed in Appendix B.1.3, the auxiliary member shapes shown in figure B.1d are obtained, representing the main load-bearing members of the part.

B.1.2 Image processing method

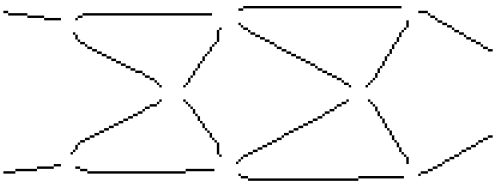
As an alternative to the stress criterion, methods from image processing can be used for an automatic subdivision of TO results. Image processing algorithms work on binary images in 2D or volumetric voxel images in 3D. While the image processing method is more general compared to the stress criterion method (e.g. since it can also handle designs obtained from a TO involving multiple load cases), it requires the use of dedicated algorithms.

For regular FE meshes with a rectangular design space, a binary image can be easily obtained by treating each finite element directly as a pixel or voxel of an image and projecting the density values with a hard threshold value of e.g. 0.5 into a binary representation. Irregular meshes or complicated design space geometries may be sampled into a suitable rectangular image of finer resolution. After processing, the results are then mapped back to the original elements.

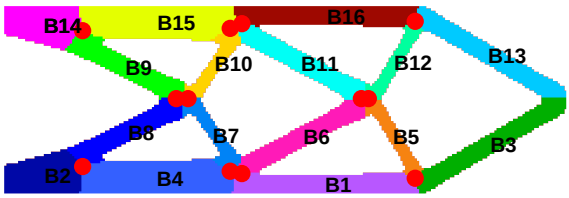
There exist more refined methods to identify trusses from TO results via image processing in the literature (see e.g. [116, 120] or [121] for a PDE-based method), which mainly tackle problems arising from a downstream CAD interpretation. This aspect is however not relevant for this work, such that the most basic image processing algorithms are sufficient to achieve the shown results.



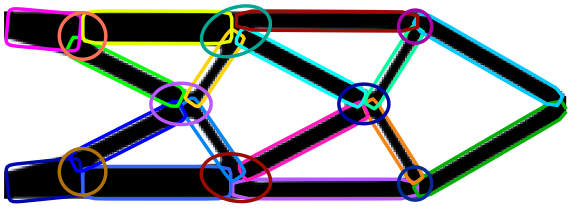
(a) Skeleton of the solid phase[25]



(b) Cut skeleton[25]



(c) Beam clusters after mapping to the skeleton and branch point locations (red)



(d) Resulting auxiliary member shapes on density field

Figure B.2 – Load path identification with image processing algorithms for the cantilever beam example.

Skeletonization: In MATLAB, the skeleton of a part's image can be extracted by the command `bwskel`. The result will be a one pixel wide center line along each feature of the part obtained by the medial axis transformation [122], as shown in figure B.2a.

The branch points of the skeleton can be extracted via the functions `bwmorph` for 2D or `bwmorph3` for 3D with the argument `'branchpoints'`. Dilated versions of these branch points will be used as the "knot" regions of the part.

The difference between the skeleton and dilated branch points gives segmented line representations of the individual "beam" regions (see figure B.2b). The region around the load introduction is disregarded since this area is usually not included in the later damage considerations.

Clustering: A mapping of each solid finite element to the nearest skeleton segment finishes the automatic classification on the "beam" side. Applied to the 2D cantilever beam example, the result from figure B.2c is obtained. The size of the "knot" regions is yet to be determined, therefore only their center points are displayed in figure B.2c.

Post-processing: The size of the "knot" regions is obtained by a simple bisection method. Assuming a circular shape, only the radius is to be determined as a size parameter per knot. Using the corresponding branch points as center points, a suitable radius for the knot is found, if the corresponding shape starts to enclose also a little amount of void from the underlying density field. Afterward, circular shapes that are very close to each other are merged into larger elliptical shapes.

Together with the size parameters for the beams, extracted according to Appendix B.1.3, the auxiliary member shapes shown in figure B.2d are obtained.

[25]The colormap of the binary image is inverted to match the representations of the pseudo densities of the TO results, i.e. black means solid and white means void, respectively.

B.1.3 Calculating parameters of auxiliary member shapes from element clusters

For each cluster, identified from the density field of a TO by one of the above-mentioned methods, the location of the center point $\boldsymbol{x}_c$, the matrix $\boldsymbol{V}$ with unit vectors of the principal axes of the auxiliary shape as well as the corresponding size vector $\boldsymbol{a}$ need to be calculated (cf. section 4.3.2). This section shows the individual calculation steps if for every cluster a set $\mathbb{S}$, containing the corresponding element indices, is given.

The center point $\boldsymbol{x}_c$ of the shape is determined by calculating the center of mass of all elements belonging to the cluster:

$$\boldsymbol{x}_c = \frac{\sum_{i\in\mathbb{S}} \boldsymbol{c}^i v_i}{\sum_{i\in\mathbb{S}} v_i} \tag{B.2}$$

The principal axes are the eigenvectors of the covariance matrix of the coordinates of all elements belonging to a cluster. The corresponding element center points $\boldsymbol{c}^i$ are concatenated into a matrix $\boldsymbol{C}$ and the covariance matrix $\boldsymbol{\Sigma}$ is calculated:

$$\boldsymbol{C} = [\boldsymbol{c}^i, \cdots] \qquad \forall i \in \mathbb{S} \tag{B.3a}$$
$$\boldsymbol{\Sigma} = \mathrm{Cov}(\boldsymbol{C}) \tag{B.3b}$$

A matrix $\boldsymbol{V}$ with unit vectors of the principal axes as columns is obtained by an eigendecomposition of $\boldsymbol{\Sigma}$:

$$\boldsymbol{\Sigma V} = \boldsymbol{V\Lambda} \tag{B.4a}$$
$$\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3] \tag{B.4b}$$
$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \tag{B.4c}$$

The eigenvalues on the diagonal of $\boldsymbol{\Lambda}$ are used to determine the size vector $\boldsymbol{a}$ with the sizes of the shape in every principal direction. Assuming that structural beams have an almost constant thickness over their length, the location of the centers of their underlying elements can be treated like uniformly distributed random coordinates. The eigenvalues of the covariance matrix would then be the squared standard deviations:

$$\sigma_i^2 = \lambda_i \tag{B.5}$$

For a 1D uniform distribution with a width of $2a$, the relation to the standard deviation σ is:

$$\sigma^2 = \frac{(2a)^2}{12} \qquad \Leftrightarrow \qquad a = \sqrt{3}\sigma \tag{B.6}$$

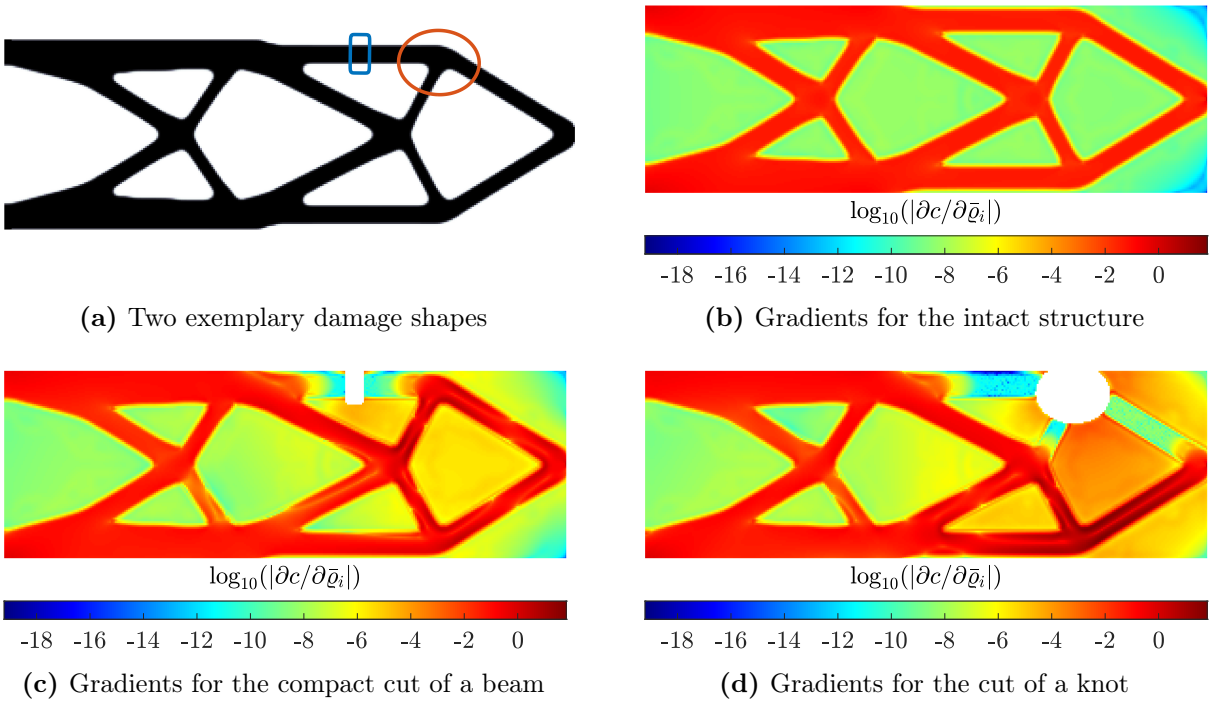
This relation can be applied to all principal directions of the shape to obtain the size vector $\boldsymbol{a}$:

$$\boldsymbol{a} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \sqrt{3} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix} \tag{B.7}$$

Applying the factor $\sqrt{3}$ from equation (B.6) as a scaling factor between the standard deviations calculated by equation (B.5) and the semi-axes of the auxiliary shapes stored in $\boldsymbol{a}$ will give a close fit of the member shapes to the original density field (cf. figures 4.14a and 4.14c).

B.2 Impact of cuts and valid simplifications for the load-path-based damage model

Figure B.3 shows the impact of two exemplary cuts at a "beam" and "knot" location on the gradients of the compliance of the optimized 2D cantilever design from figure 2.8c. Therein, each element i is colored according to the sensitivity of the part's compliance with respect to the element's projected density, which is $\frac{\partial c}{\partial \bar{\varrho}_i}$. Since the gradient for compliance has only negative entries, the absolute value is taken such that a logarithmic scale can be used. The gradients inside the damage zone are zero (cf. Appendix D.8), therefore they tend to $-\infty$ and are disregarded in figures B.3c and B.3d.



(a) Two exemplary damage shapes

(b) Gradients for the intact structure

(c) Gradients for the compact cut of a beam

(d) Gradients for the cut of a knot

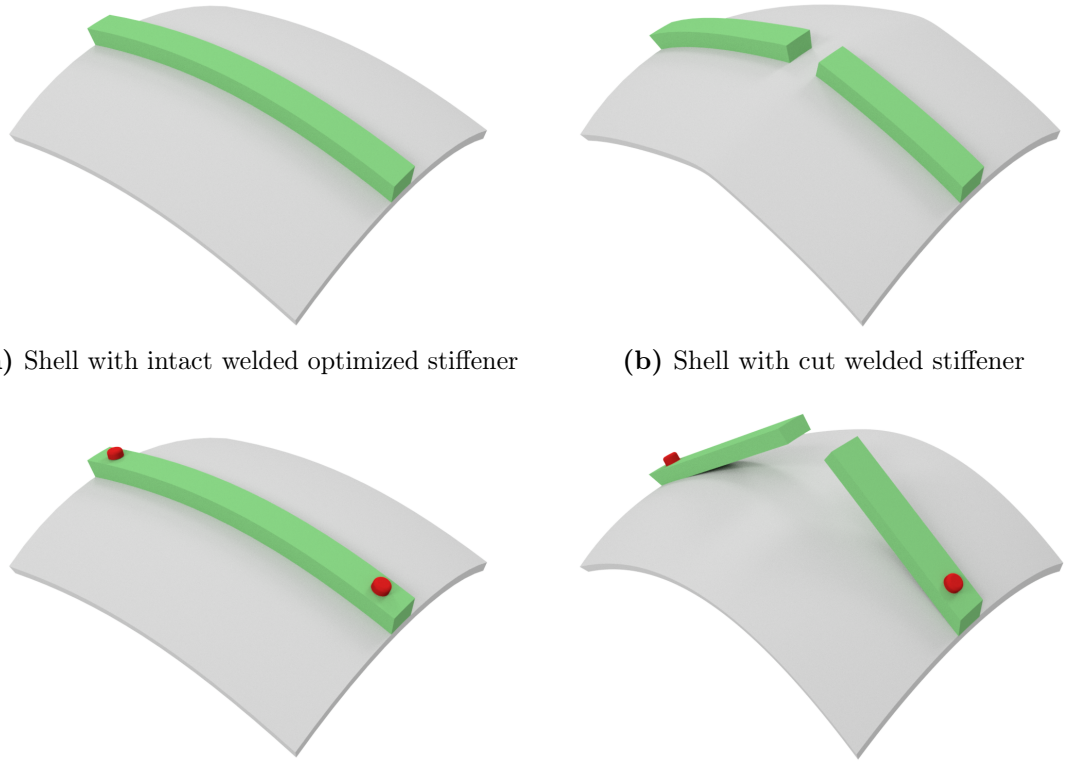
Figure B.3 – Absolute values of the gradients on a logarithmic scale for the intact and cut cantilever beam designs.

For the studied examples in this work, involving compliance as objective, cutting a beam in the middle is sufficient to remove its capability to carry loads. As shown in figure B.3c, the magnitude of the gradient for the whole beam then drops by many orders to a level similar to pure void zones for the intact structure shown in figure B.3b. This property also underlines, that in the shown examples the proposed load-path-based damage model is not sensitive to the exact placement of damage zones, as long as they fully cut through their corresponding structural member.

Also, for compliance as objective, removing material at a knot region affects all beams intersecting at this position and thus is expected to have a more severe impact, than a cut of a single beam. This is confirmed by figure B.3d, where it is clearly visible, that a cut of a knot affects the compliance gradients in all three adjacent beams.

Though both above-mentioned findings are valid for the shown examples in this work, they cannot be generalized for different types of optimizations. For instance, locally disconnected beams may lead to undesirable vibration modes in fail-safe optimizations involving dynamic analyses [123]. In this case, the extent of a cut applied to a beam makes a difference and a compact cut through the middle of a beam, like the one shown in figure B.3a, may be a suboptimal choice, since then two free ends remain at the structure. Also, if e.g. stress is considered as objective in the optimization, even a partial degradation of structural members may be worse than a full removal [124] and the removal of one member may be worse than the removal of several members at once [33], at least when evaluated in a linear analysis. In these cases, the assumption, that damage on a knot region has a larger impact than a damage on single beams is not conservative anymore.

Another important aspect when considering simplifications in the damage model, even for compliance-based problems, is the presence of solid non-design spaces (NDS) that may introduce load-bearing "parallel" structures to the design space (DS) being optimized. Figure B.4 shows such an example, where a loaded shell is present as NDS on which a stiffener layout is to be optimized via the proposed methods in this work. The behavior under damage of the joint structure will then depend on the model of the interaction between the DS and NDS. Two exemplary interaction types are discussed in the following.



(a) Shell with intact welded optimized stiffener

(b) Shell with cut welded stiffener

(c) Shell with intact riveted optimized stiffener

(d) Shell with cut riveted stiffener

Figure B.4 – A loaded shell as NDS with an optimized stiffener on top. Depending on the connection type, a cut in the middle might not be sufficient to model a complete loss of the stiffening property.

In the case shown in figures B.4a and B.4b, the stiffener is considered as welded on top of the shell, such that force transfer is possible over the whole contact surface. A small cut through the middle of the stiffener is thus not equivalent to a complete loss of the whole stiffener, since

the shell is able to bridge the force transfer over the introduced gap and only a very localized stiffness degradation is achieved. In fact, also locally disconnected stiffeners contribute to a minimization of the overall compliance of the structure and are therefore part of such topology optimized stiffener layouts on shells (see e.g. [21]). Similar thoughts can be made concerning knot locations if multiple intersecting stiffeners are present. Failure of the knot would then not correspond to a failure of all attached stiffeners.

However, if the connection is modeled via point-wise connectors like bolts or rivets, as shown in figures B.4c and B.4d, and no force transfer is possible in between the connectors, the bridging effect is not present and above-mentioned problems are irrelevant. This allows to go back to the compact damage cuts, that are also used throughout the examples in this work.

B.3 Distribution of bending moments under damaged conditions

To investigate the influence of the bending moments on the results shown in figure 5.5 of section 5.3.1, an additional FE model of the preliminary geometry from figure 5.5a is set up. Therein, the structure is discretized by beam finite elements where each beam has internally a width parameter assigned to it that is extracted from the original density field.

In figure B.5, the results of the FE analyses, showing the absolute values of the cross-sectional bending moments as a line representation with scaled displacements, are shown. For the intact structure from figure B.5a, the bending moments are relatively low, the load is mostly carried by tension and compression forces. However, if one of the beams fails, the bending moments become dominant, as demonstrated in the other figures B.5b – B.5h, which show the bending moments for cutting beams in the upper region of the part from left to right.

Under damaged conditions, the inner loads are redistributed such that in many beams the function of the bending moment is zero in the center, which explains the shape of the beams of the optimized design shown in figure 5.5c.

B.4 Damage shape scaling and circumvention effect

After the preliminary design is obtained in optimization stage I, auxiliary shapes are calculated for every beam and knot region of the structure. The member shapes will fit very closely to the original structure (cf. figure 4.14).

If the damage shapes derived from the member shapes (see section 4.3.4) are not scaled up, they may easily be circumvented during the fail-safe SO in stage II. Considering the example involving the SO of the "robust" cantilever beam from section 5.3.1, without up-scaling the final result from figure B.6a is obtained which clearly differs from the presented optimized result shown in figure 5.5c. The most critical damage shapes are circumvented by material that is placed just around the damage shape, hinders the corresponding structural member to be cut entirely, and therefore jeopardizes the fail-safe optimization.

The worst-case compliance $\bar{c}$ for damages with a damage shape scaling factor of 1 is given in table B.1 and is now much lower than for the example in section 5.3.1 that uses a scaling factor of 2. This is not due to an increased damage tolerance of the structure but rather due to the incomplete cutting of the individual load paths. Figure B.6b shows some exemplary incomplete cuts in the upper region of the part.

The other extreme would be a much too high damage scaling factor. The damage shapes and the result after a fail-safe SO for a scaling factor of 3.5 are shown in figure B.7a[26]. Because these

[26] An even larger scaling factor of e.g. 4 would yield a damage shape on the central front knot that is so large, that it completely disjoins the whole design space, such that no connection of the load to the BC region is possible.

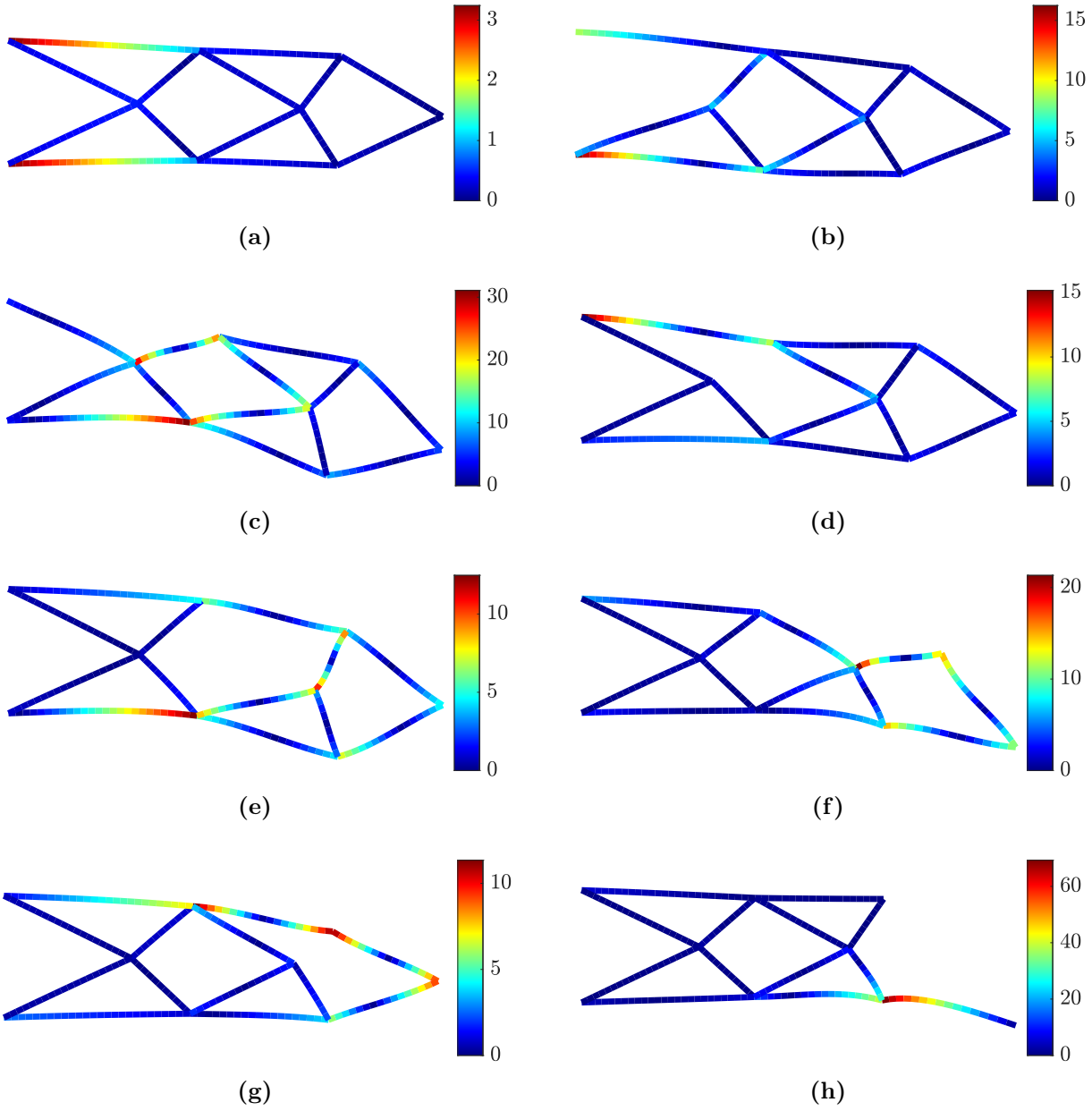
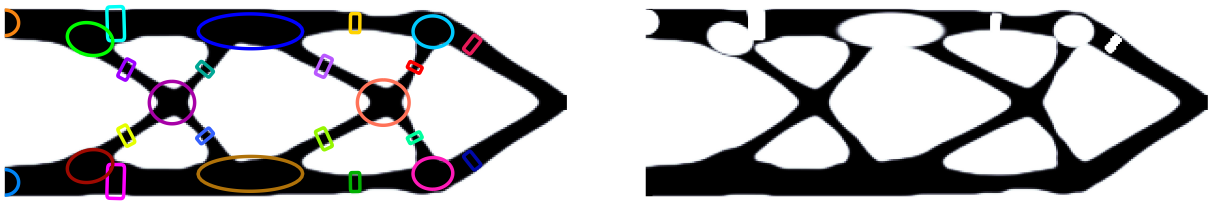


Figure B.5 – Absolute values for the cross-sectional bending moments for a beam FE model of the "robust" cantilever design.

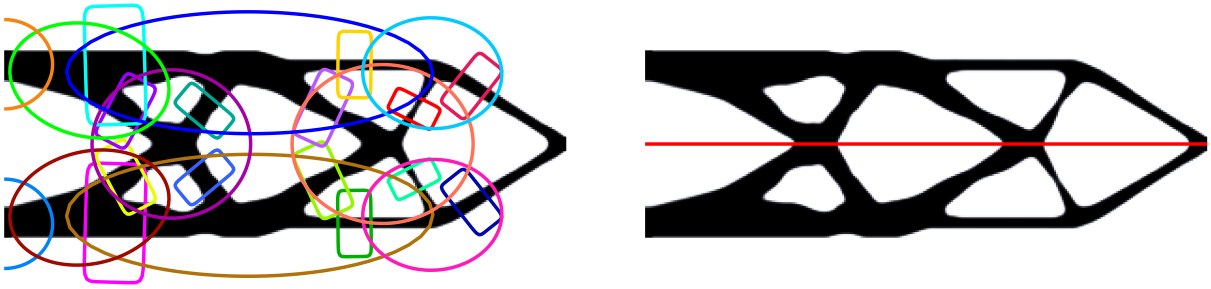


(a) Outlines of damage shapes on optimized design

(b) Some incomplete cuts at potentially critical locations at the top shown simultaneously

Figure B.6 – Results of the fail-safe SO for a damage shape scaling of 1.

overly large damage regions are an even more conservative approach than the ones with the recommended scaling factor of 2 (shown in figure 5.5b), also the worst-case compliance $\bar{c}$ listed in table B.1 has increased, but only slightly. The difference in the final structure is minimal when comparing the scaling factors 3.5 and 2, as shown in figure B.7b.



(a) Outlines of damage shapes on optimized design

(b) Optimized design for a scaling factor of 3.5 (top) and 2 (bottom)

Figure B.7 – Results of the fail-safe optimization for a damage shape scaling of 3.5.

Table B.1 – Worst-case compliances $\bar{c}$ for optimizations with different damage scaling factors.

Damage scaling factor	Figure	$\bar{c}$
1	B.6a	629.92
2	5.5c	7 364.49
3.5	B.7a	7 387.79

B.5 Convergence plots for density-based shape optimization

Convergence plots for the shape optimized fail-safe 2D examples from section 5.3 are shown in figure B.8. The function values are normalized with respect to the values at the initial stage.

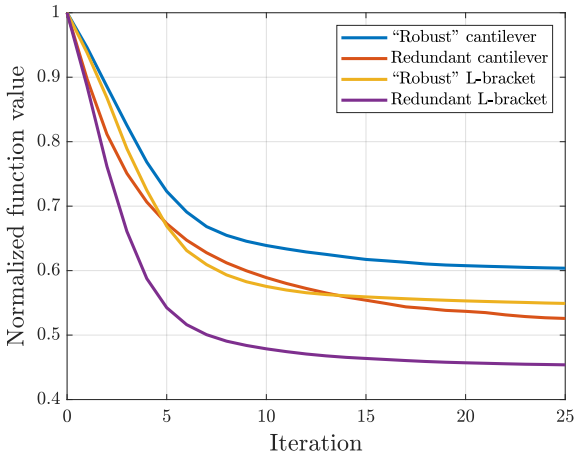


Figure B.8 – Optimization history of objective values for the proposed density-based SO method.

Fast convergence is observed and most of the improvement is already achieved within the first 10 iterations. In total, 25 iterations are performed for each shape optimized example presented in this work.

Note, that the curves shown in figure B.8 refer to the aggregated values over all damage cases considered in the optimizations and thus show slight differences to the real worst-case values given in the respective tables in section 5.3.

C Modeling aspects in the three field topology optimization framework

This appendix focuses on additional remarks, modeling aspects and settings related to the three field TO framework used in this work.

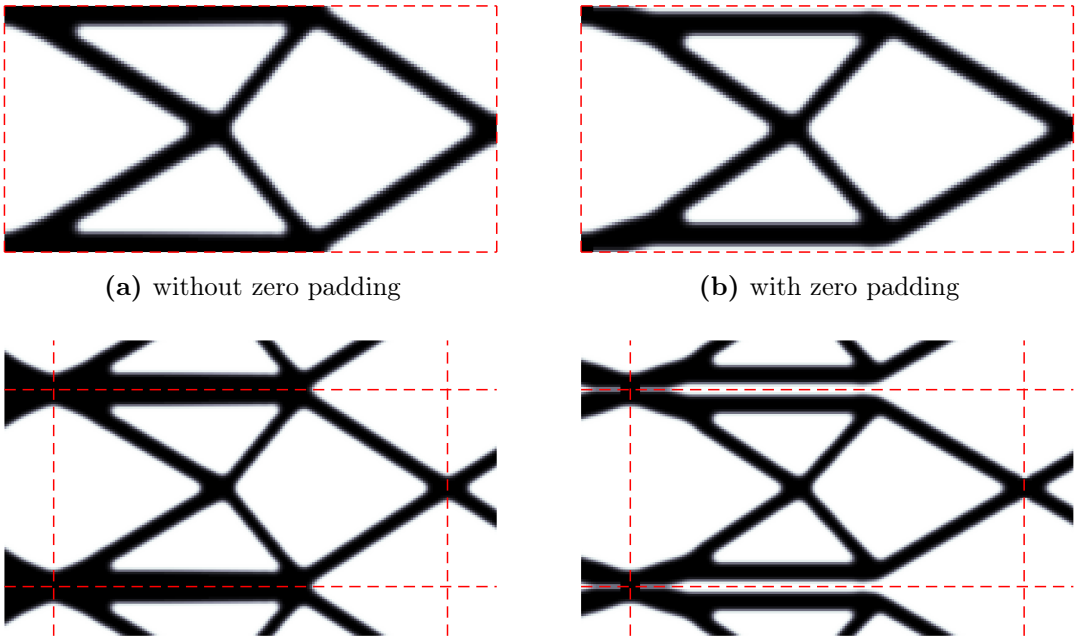
C.1 Zero padding at the filtering stage

Boundary effects are introduced by using the filter equation (2.2). Considering a regular FE mesh with a constant element size, elements at the boundaries have fewer neighboring elements within the filter radius r, than elements inside the design space that have the full neighborhood. The weighted sum from equation (2.2a) therefore includes fewer terms, such that a solid feature in the filtered field can be obtained with fewer DVs taking on the value of 1. The optimizer then tends to prefer placing solid material at boundary locations, such that a "sticking" effect (see e.g. [66] and figure C.1) is observed, which is especially problematic for the density-based shape optimization approach presented in section 5 of this work.

A short discussion of boundary treatment is given in [65], more details are given in [66], where a domain extension approach is presented to counter boundary effects. In all examples involving regular FE meshes shown in this paper, a simplified approach of "zero padding" is used as a computationally cheap way to suppress artificial symmetry effects at free boundaries of the design space. For a regular mesh and a full neighborhood, the denominator of equation (2.2a) is constant and needs to be calculated only once. Applying this value as a denominator also for elements at free boundaries effectively means, that the missing elements in their neighborhood are reintroduced with a value of 0 for their DVs. Then, the solid features gradually decrease in density towards the free boundaries. This approach is not applied for elements closer than the filter radius r to BC or loading regions, in order to allow these regions to become fully solid even at the boundary.

Applying the above-mentioned simplified zero padding approach also for irregular meshes is not straightforward, however, methods like the "appoximate volume approach" from [125] exist as a means to adapt the basic idea also for irregular meshes.

Figure C.1 exemplary shows results obtained for a minimum compliance optimization of a short cantilever beam inside a rectangular design domain using the unmodified filtering approach (figure C.1a) and using zero padding (figure C.1b). By expanding the structures with their mirrored counterparts at the boundaries of the design domain, it is clearly visible, that without zero padding an artificial symmetry is enforced, with material edges entering the design domain at a right angle, as seen in figure C.1c. However, when using zero padding, neither the aforementioned sticking nor perpendicular reentry angles are observed at the traction-free boundaries in figure C.1d.



(a) without zero padding **(b)** with zero padding

(c) without zero padding and mirrored **(d)** with zero padding and mirrored

Figure C.1 – Difference between standard filtering and filtering with zero padding.

C.2 Effect of material penalization, filtering, and projection

Figure C.2 demonstrates, how the different parameters influence the result of a TO. The objective is minimization of compliance with a volume constraint for a cantilever beam that is loaded with a transverse force at the tip.

If no material penalization is used (meaning, the stiffness of an element scales linearly with the pseudo density, as with $p = 1$ in SIMP or $q = 0$ in RAMP), large areas of intermediate densities are observed, as shown in figure C.2a.

Activating penalization with SIMP and $p = 3$ (see section 2.3.4) without a filtering step gives the result seen in figure C.2b, where areas of checkerboard patterns can be observed. Checkerboard patterns form an (undesired) local optimum due to a numerically induced, artificially high stiffness of the corresponding material layout (for a discussion, see [27, 126]).

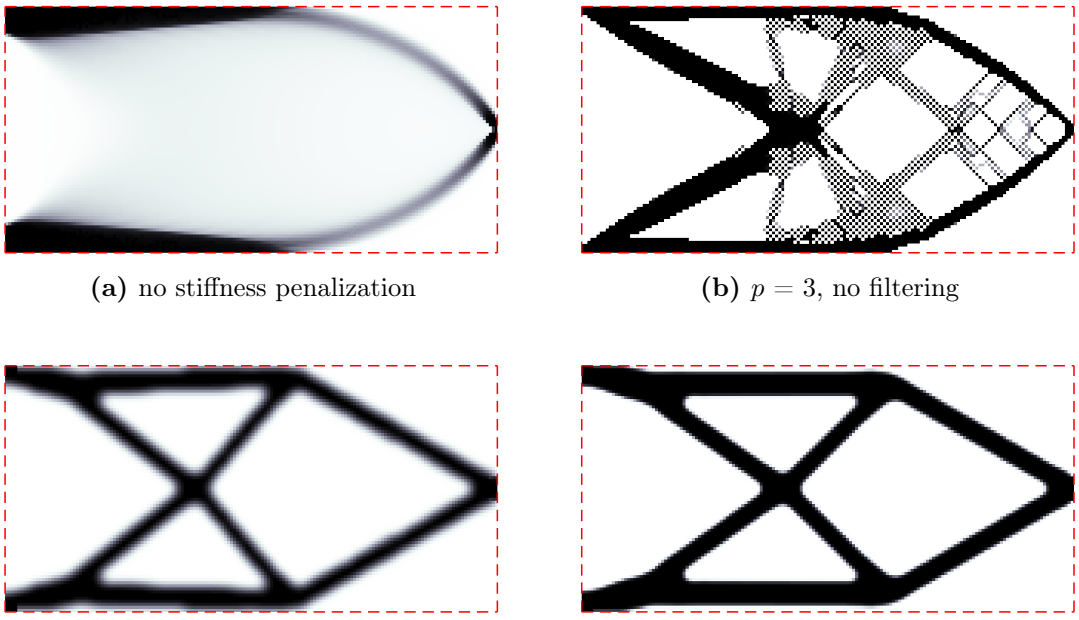
Using a filtering step (see section 2.3.1) suppresses the checkerboard patterns, as seen in figure C.2c. The result is now dominated by the chosen filter radius r and not the resolution of the underlying FE mesh. However, due to the averaging performed in the filtering step, the outlines of the part are blurred.

To make the transition regions from solid to void more compact, the projection step discussed in section 2.3.2 is used and the result from figure C.2d is obtained.

C.3 Effect of a uniform variation of η

For the "robust" approach [31], eroded and dilated realizations of a part are considered besides the nominal "blueprint" part. The eroded and dilated designs are obtained by changing the projection threshold value η to model uniform manufacturing errors.

Assuming $\eta = 0.5$ as baseline value for the nominal design, lowering the value of η means that filtered variables with intermediate values $\tilde{\varrho}_i < 0.5$ are already projected towards the solid phase (cf. the curves in figure 2.4b). This effectively thickens all structural members of the part,



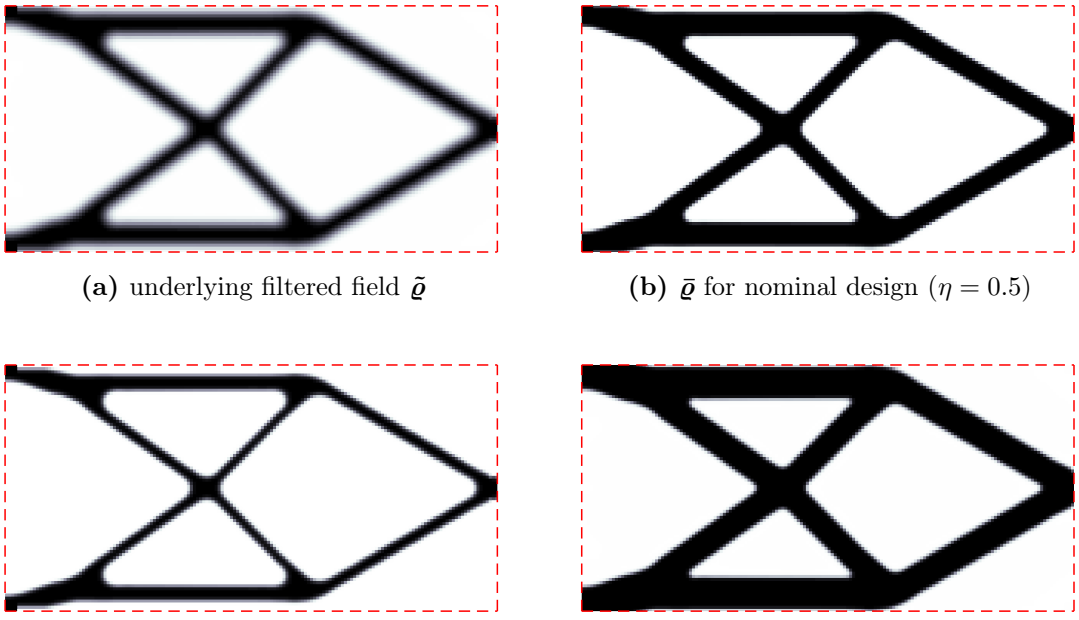
(a) no stiffness penalization

(b) $p = 3$, no filtering

(c) $p = 3$, filtering without projection

(d) $p = 3$, filtering and projection

Figure C.2 – Cantilever beam results when activating material penalization, filtering, and projection.



(a) underlying filtered field $\tilde{\boldsymbol{\varrho}}$

(b) $\bar{\boldsymbol{\varrho}}$ for nominal design ($\eta = 0.5$)

(c) $\bar{\boldsymbol{\varrho}}$ for eroded design ($\eta = 0.8$)

(d) $\bar{\boldsymbol{\varrho}}$ for dilated design ($\eta = 0.2$)

Figure C.3 – Cantilever beam results with the three realizations of nominal, eroded, and dilated designs.

yielding a dilated design. The opposite effect of erosion, i.e. thinning of all structural members, is achieved by increasing the threshold η.

Figure C.3 shows the underlying filtered field, as well as the three realizations of the projected field for the nominal, eroded, and dilated designs for a compliance minimization of a short cantilever beam.

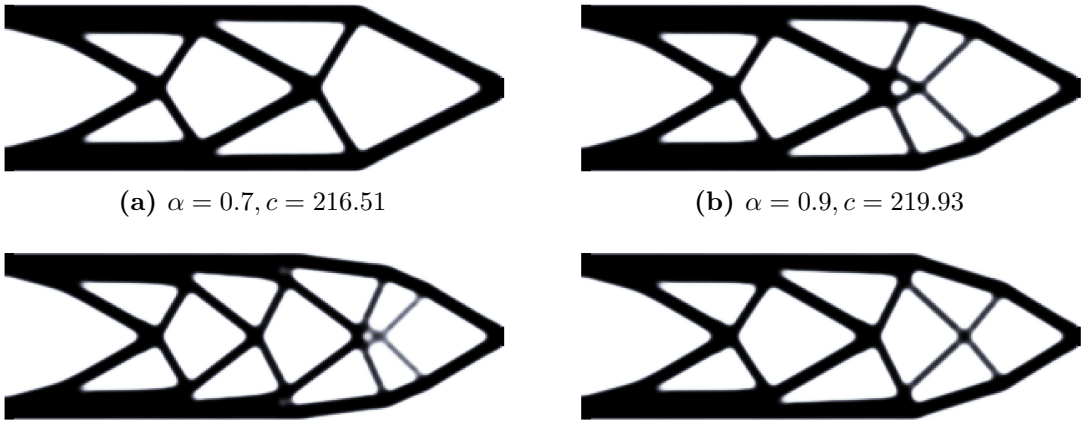
C.4 Effect of using different uniform starting fields

Figure C.4 shows four results obtained for a compliance minimization of a cantilever beam under a global volume constraint using the plain three field TO approach from section 2.3.

The domain is discretized by 300×100 elements, the filter radius is set to $r = 6$ (which is larger than the radius used for the same problem in section 2.7.1), and the global volume fraction is restricted to a value of $k_g = 0.4$. Other parameters are listed in table C.1. The only difference between the optimizations is the choice of the starting values for the DVs, which are initialized by a uniform field with a value of k_g scaled by a prefactor α:

$$\boldsymbol{\varrho}_{initial} = \alpha k_g \boldsymbol{I} \tag{C.1}$$

In figure C.4, the results for four exemplary chosen α values are shown, together with the final compliances of the designs. The results are local minima of the optimization problem. They differ in the layout of the structural members and three different topologies are obtained. Interestingly, the result from figure C.4a with the simplest topology has the lowest compliance in this case.



(a) $\alpha = 0.7, c = 216.51$ **(b)** $\alpha = 0.9, c = 219.93$

(c) $\alpha = 1.0, c = 221.67$ **(d)** $\alpha = 1.1, c = 219.84$

Figure C.4 – Optimized cantilever beams and their compliances c obtained with different uniform starting fields.

C.5 Element types used

For all shown examples in this work involving regular FE meshes, linear, fully integrated, isoparametric finite elements are used (c.f. [127]). The local coordinates of these elements are:

$$\boldsymbol{\xi} = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} \text{ for 2D}, \qquad \boldsymbol{\xi} = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} \text{ for 3D}, \qquad -1 \leq \xi_i \leq 1 \tag{C.2}$$

For the 2D case, four-node quadriliteral elements are used that are integrated assuming a plane stress state. The ansatz functions are:

$$N_1 = \frac{1}{4}(1+\xi_1)(1+\xi_2) \quad \text{(C.3a)} \qquad N_3 = \frac{1}{4}(1-\xi_1)(1-\xi_2) \quad \text{(C.3c)}$$

$$N_2 = \frac{1}{4}(1-\xi_1)(1+\xi_2) \quad \text{(C.3b)} \qquad N_4 = \frac{1}{4}(1+\xi_1)(1-\xi_2) \quad \text{(C.3d)}$$

For the 3D case, eight-node hexahedral element with following ansatz functions are used:

$$N_1 = \frac{1}{8}(1+\xi_1)(1+\xi_2)(1+\xi_3) \quad \text{(C.4a)} \qquad N_5 = \frac{1}{8}(1+\xi_1)(1+\xi_2)(1-\xi_3) \quad \text{(C.4e)}$$

$$N_2 = \frac{1}{8}(1-\xi_1)(1+\xi_2)(1+\xi_3) \quad \text{(C.4b)} \qquad N_6 = \frac{1}{8}(1-\xi_1)(1+\xi_2)(1-\xi_3) \quad \text{(C.4f)}$$

$$N_3 = \frac{1}{8}(1-\xi_1)(1-\xi_2)(1+\xi_3) \quad \text{(C.4c)} \qquad N_7 = \frac{1}{8}(1-\xi_1)(1-\xi_2)(1-\xi_3) \quad \text{(C.4g)}$$

$$N_4 = \frac{1}{8}(1+\xi_1)(1-\xi_2)(1+\xi_3) \quad \text{(C.4d)} \qquad N_8 = \frac{1}{8}(1+\xi_1)(1-\xi_2)(1-\xi_3) \quad \text{(C.4h)}$$

C.6 Parameters for the shown academic examples

The parameters listed in table C.1 are used for the shown academic examples involving regular FE meshes.

Table C.1 – Optimization parameters used for the academic examples.

Parameter	Value
Element size in 2D	1×1
Element size in 3D	$1 \times 1 \times 1$
Domain size	problem-specific
Filter radius r	problem-specific
Projection β initial value	2
Projection β maximum value	8
Projection β continuation	$\{2, 4, 8\}$ at iter. $\{0, 50, 100\}$
Projection η for standard TO	0.5
Material interpolation scheme	SIMP
SIMP penalty exponent p	3
Young's modulus solid phase E_0	1
Young's modulus void phase E_{min}	10^{-9}
Poisson's ratio ν	0.3
Maximum iterations	200
Convergence tolerance	10^{-6}

D Gradients for plain and fail-safe topology optimization

This appendix lists additional derivatives required for approaching the presented TO (and SO) methods of this work by gradient-based optimization algorithms.

D.1 Sensitivities for the three field topology optimization

For the standard three field topology optimization framework discussed in section 2.3, the projected densities $\bar{\boldsymbol{\varrho}}$ are the "physical" densities that are considered for assembling the system's stiffness matrix.

Sensitivities, that are obtained with relation to the projected densities can be transformed to sensitivities with relation to the original design variables $\boldsymbol{\varrho}$ by applying the chain rule:

$$\frac{d(\cdot)}{d\varrho_i} = \cancelto{0}{\frac{\partial(\cdot)}{\partial\varrho_i}} + \sum_j \frac{\partial(\cdot)}{\partial\bar{\varrho}_j}\frac{\partial\bar{\varrho}_j}{\partial\tilde{\varrho}_j}\frac{\partial\tilde{\varrho}_j}{\partial\varrho_i} \tag{D.1}$$

For plain topology optimization problems, the direct partial derivative as first term in equation (D.1) is zero for all objectives and constraints considered in this work. The remaining partial derivatives of the filter and projection equations are given in Appendix D.3 and D.4.

D.2 Adjoint sensitivity analysis for objective functions

Adjoint sensitivity analysis is used to obtain the derivatives of objective or constraint functions that are dependent on the displacement vector $\boldsymbol{u}$. In the following, the exemplary case of an objective function $\Phi(\boldsymbol{u})$ is studied.

For gradient-based optimization, the sensitivities of the objective function with relation to the DVs are needed. The first step is therefore to obtain the derivative with respect to the projected densities (or modified densities, where applicable), as explained in Appendix D.1.

However, simply taking the total derivative with respect to the projected densities would yield:

$$\frac{d\Phi}{d\bar{\varrho}_i} = \cancelto{0}{\frac{\partial\Phi}{\partial\bar{\varrho}_i}} + \frac{\partial\Phi}{\partial\boldsymbol{u}}\frac{\partial\boldsymbol{u}}{\partial\bar{\varrho}_i} \tag{D.2}$$

In practice, obtaining the matrix of all partial derivatives $\frac{\partial\boldsymbol{u}}{\partial\bar{\varrho}_i}$ is not only computationally extremely expensive, additionally, for a TO problem, this matrix is also fully populated. With a size of $n_{DOF} \times n_e$, already medium-sized problems require an unfeasible amount of memory to store this gradient matrix.

Adjoint sensitivity analysis eliminates the need to calculate the derivatives $\frac{\partial\boldsymbol{u}}{\partial\bar{\varrho}_i}$ [63, 128]. To achieve this, the state equation $\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}$ is incorporated as an equality constraint into the objective function by augmenting Φ with the residual, pre-multiplied by Lagrange multipliers $\boldsymbol{\lambda}$:

$$\Phi = \Phi + \boldsymbol{\lambda}^T(\boldsymbol{K}\boldsymbol{u} - \boldsymbol{f}) \tag{D.3}$$

If the displacement vector $\boldsymbol{u}$ is a solution to the state equation $\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}$, i.e. the residual is a vector of zeros, the choice of $\boldsymbol{\lambda}$ in equation (D.3) is arbitrary and the augmented objective function takes on the same value as the original objective function.

Assuming, that neither the objective function Φ nor the external force vector $\boldsymbol{f}$ depend on the projected densities, the derivative can be obtained by:

$$\frac{d\Phi}{d\bar{\varrho}_i} = \cancelto{0}{\frac{\partial\Phi}{\partial\bar{\varrho}_i}} + \frac{\partial\Phi}{\partial\boldsymbol{u}}\frac{\partial\boldsymbol{u}}{\partial\bar{\varrho}_i} + \boldsymbol{\lambda}^{\boldsymbol{T}}\left(\frac{\partial\boldsymbol{K}}{\partial\bar{\varrho}_i}\boldsymbol{u} + \boldsymbol{K}\frac{\partial\boldsymbol{u}}{\partial\bar{\varrho}_i} - \cancelto{0}{\frac{\partial\boldsymbol{f}}{\partial\bar{\varrho}_i}}\right) \tag{D.4a}$$

$$= \left(\frac{\partial\Phi}{\partial\boldsymbol{u}} + \boldsymbol{\lambda}^T\boldsymbol{K}\right)\frac{\partial\boldsymbol{u}}{\partial\bar{\varrho}_i} + \boldsymbol{\lambda}^T\frac{\partial\boldsymbol{K}}{\partial\bar{\varrho}_i}\boldsymbol{u} \tag{D.4b}$$

The adjoint problem, eliminating the first term in equation (D.4b), states (for a symmetric $\boldsymbol{K}$):

$$\boldsymbol{K}^T\boldsymbol{\lambda} = \boldsymbol{K}\boldsymbol{\lambda} = -\frac{\partial\Phi}{\partial\boldsymbol{u}}^T \tag{D.5}$$

Having obtained the Lagrange parameters $\boldsymbol{\lambda}$ by equation (D.5), the derivative of the objective function is the remaining second term of equation (D.4b):

$$\frac{d\Phi}{d\bar{\varrho}_i} = \boldsymbol{\lambda}^T\frac{\partial\boldsymbol{K}}{\partial\bar{\varrho}_i}\boldsymbol{u} \tag{D.6}$$

The computational cost of the adjoint method equals solving the system matrix $\boldsymbol{K}$ for one additional right-hand side (or pseudo load vector) per entry of Φ, compared to one pseudo load vector for every DV in the direct approach.

The solution to the pseudo load vector in the adjoint method can be obtained simultaneously with solving for the displacements $\boldsymbol{u}$ as response to the real external force vector $\boldsymbol{f}$. For cases, where assembling the pseudo load vector requires $\boldsymbol{u}$ to be already available, a decomposition of $\boldsymbol{K}$ can be reused, making this approach still very fast and efficient. In the special case of compliance as objective function, the pseudo load vector equals the negative applied mechanical load, such that for linear systems no additional solve of the system matrix is needed (see section 2.4.1).

D.3 Derivative of the filter equation

The derivatives of the filtered variables $\tilde{\boldsymbol{\varrho}}$ with respect to the DVs $\boldsymbol{\varrho}$ are:

$$\frac{\partial\tilde{\varrho}_i}{\partial\varrho_j} = \frac{w(\boldsymbol{c}^j - \boldsymbol{c}^i)v_j}{\sum_j w(\boldsymbol{c}^j - \boldsymbol{c}^i)v_j} \tag{D.7}$$

D.4 Derivatives of the projection equation

Here, the derivatives of the projected densities $\bar{\boldsymbol{\varrho}}$ with respect to the filtered variables $\tilde{\boldsymbol{\varrho}}$ and the threshold parameter η are given. Following abbreviations are used:

$$A := \tanh(\beta\eta), \quad B := \tanh\left(\beta(\tilde{\varrho}_i - \eta)\right), \quad C := \tanh\left(\beta(1-\eta)\right) :$$

$$\frac{\partial \bar{\varrho}_i}{\partial \tilde{\varrho}_i} = \frac{\beta \operatorname{sech}^2\left(\beta(\tilde{\varrho}_i - \eta)\right)}{A + C} \tag{D.8}$$

$$\frac{\partial \bar{\varrho}_i}{\partial \eta} = \frac{\beta\left(\operatorname{sech}^2(\beta\eta)(C - B) - \operatorname{sech}^2\left(\beta(\tilde{\varrho}_i - \eta)\right)(A + C) + \operatorname{sech}^2\left(\beta(1 - \eta)\right)(A + B)\right)}{(A + C)^2} \tag{D.9}$$

In cases, where the threshold parameter is defined per element (i.e. as a threshold vector $\boldsymbol{\eta}$), the variable η in equation (D.9) is replaced by the corresponding elemental threshold value η_i, yielding the derivative $\frac{\partial \bar{\varrho}_i}{\partial \eta_i}$.

D.5 Sensitivities of the global stiffness matrix for plain TO

For the standard TO approach, where every finite element is assigned an own projected variable $\bar{\varrho}_i$, the derivative of the global stiffness matrix $\boldsymbol{K}$ with respect to the projected variable is simply the derivative of the i-th element's stiffness matrix sorted into the global system:

$$\frac{\partial \boldsymbol{K}}{\partial \bar{\varrho}_i} = \mathrm{A}\, \frac{\partial \boldsymbol{k}_i}{\partial \bar{\varrho}_i} \tag{D.10}$$

Here, "A" is the assembly operator, this time sorting only a single element's derivative matrix into the global system.

For the SIMP approach and $p \geq 1$, the derivative of the element's matrix $\boldsymbol{k}_i$ with respect to the projected variables is:

$$\frac{\partial \boldsymbol{k}_i}{\partial \bar{\varrho}_i} = \left(p \bar{\varrho}_i^{p-1}(E_0 - E_{min})\right) \boldsymbol{k}_0 \tag{D.11}$$

And for the RAMP approach with $q \geq 0$:

$$\frac{\partial \boldsymbol{k}_i}{\partial \bar{\varrho}_i} = \left(\frac{1 + q}{\left(1 + q(1 - \bar{\varrho}_i)\right)^2}(E_0 - E_{min})\right) \boldsymbol{k}_0 \tag{D.12}$$

The unitary elemental stiffness matrix $\boldsymbol{k}_0$ is obtained by using a Young's modulus of $E = 1$.

D.6 Derivatives of aggregation functions

The gradients of the studied aggregation functions with respect to the DVs ϱ_i are as follows:

- For the p-norm aggregation function A_{pn}:

$$\frac{dA_{pn}(\boldsymbol{\tau})}{d\varrho_i} = \sum_{j=1}^{n_t} \frac{\partial A_{pn}}{\partial \tau_j} \frac{\partial \tau_j}{\partial \varrho_i} = \sum_{j=1}^{n_t} \underbrace{\left(\sum_{k=1}^{n_t} \tau_k^p\right)^{\frac{1}{p}-1} \tau_j^{p-1}}_{w_j} \frac{\partial \tau_j}{\partial \varrho_i} \tag{D.13}$$

- For the p-mean aggregation funtion A_{pm}, since $A_{pm}(\boldsymbol{\tau}) = \left(\frac{1}{n_t}\right)^{\frac{1}{p}} A_{pn}(\boldsymbol{\tau})$:

$$\frac{dA_{pm}(\boldsymbol{\tau})}{d\varrho_i} = \left(\frac{1}{n_t}\right)^{\frac{1}{p}} \frac{dA_{pn}(\boldsymbol{\tau})}{d\varrho_i} = \sum_{j=1}^{n_t} \underbrace{\left(\frac{1}{n_t}\right)^{\frac{1}{p}} \left(\sum_{k=1}^{n_t} \tau_k^p\right)^{\frac{1}{p}-1} \tau_j^{p-1}}_{w_j} \frac{\partial \tau_j}{\partial \varrho_i} \tag{D.14}$$

- For the KS aggregation function A_{KS}:

$$\frac{dA_{KS}(\boldsymbol{\tau})}{d\varrho_i} = \sum_{j=1}^{n_t} \frac{\partial A_{KS}}{\partial \tau_j} \frac{\partial \tau_j}{\partial \varrho_i} = \sum_{j=1}^{n_t} \underbrace{\frac{\mathrm{e}^{p\tau_j}}{\sum_{k=1}^{n_t} \mathrm{e}^{p\tau_k}}}_{w_j} \frac{\partial \tau_j}{\partial \varrho_i} \tag{D.15}$$

D.7 Sensitivities for the four field TO considering failure

For optimization problems considering also damaged configurations of a structure, "modified densities" $\hat{\boldsymbol{\varrho}}$ are the physical densities used in the FE calculation. The modified densities are derived from the intact configuration of the structure that is expressed by the projected densities $\bar{\boldsymbol{\varrho}}$.

The sensitivities with relation to the original design variables $\boldsymbol{\varrho}$ are obtained by applying the chain rule. The chain length is now four since in contrast to the plain three field TO approach, with its sensitivities derived in Appendix D.1, the additional field of modified densities exists:

$$\frac{d(\cdot)}{d\varrho_i} = \cancelto{0}{\frac{\partial(\cdot)}{\partial \varrho_i}} + \sum_j \frac{\partial(\cdot)}{\partial \hat{\varrho}_j} \frac{\partial \hat{\varrho}_j}{\partial \bar{\varrho}_j} \frac{\partial \bar{\varrho}_j}{\partial \tilde{\varrho}_j} \frac{\partial \tilde{\varrho}_j}{\partial \varrho_i} \tag{D.16}$$

For the partial derivatives of the filter and projection equations refer to Appendix D.3 and D.4. The partial derivatives $\frac{\partial \hat{\varrho}_j}{\partial \bar{\varrho}_j}$ of the modified densities depend on the employed damage model and are given in Appendix D.8.

D.8 Partial derivatives of modified densities

For the simplified damage model with rectangular damage zones on a regular mesh (see section 4.2.1 and [35]), the partial derivatives for the modified densities are:

$$\frac{\partial \hat{\varrho}_{d,i}}{\partial \bar{\varrho}_i} = \begin{cases} 1, & \text{if element } i \notin {}^{-}\Omega_d \\ 0, & \text{if element } i \in {}^{-}\Omega_d \end{cases} \tag{D.17}$$

When using a multiplicative damage mask vector (see equation (4.10a)), the partial derivative simply corresponds to the value of the mask vector for the specific element:

$$\frac{\partial \hat{\varrho}_{d,i}}{\partial \bar{\varrho}_i} = {}^{-}\psi_{d,i} \tag{D.18}$$

If material for an element i is completely removed for a damage scenario (${}^{-}\psi_{d,i} = 0$), then also the derivative of the studied function will be zero for this element and this damage case because any change in material at this location will be erased anyway (cf. equation (D.16)).

E Gradients for assembly optimization

This appendix lists derivatives relevant for the implementation of the assembly optimization methods presented in sections 6 and 7 of this work.

E.1 Partial derivatives for a single mask

The entries of a single mask vector $\boldsymbol{\psi}^i$ for a circular feature are obtained via equation (6.11). The derivatives with respect to the used reference point coordinates are:

$$\frac{\partial \psi_j^i}{\partial x^i} = -\frac{\alpha x^i}{r_m^2} \operatorname{sech}^2\left((\alpha E(\boldsymbol{c}^j - \boldsymbol{x}^i)\right) \tag{E.1a}$$

$$\frac{\partial \psi_j^i}{\partial y^i} = -\frac{\alpha y^i}{r_m^2} \operatorname{sech}^2\left((\alpha E(\boldsymbol{c}^j - \boldsymbol{x}^i)\right) \tag{E.1b}$$

E.2 Derivatives of the combined mask

The derivatives of a combined mask $\boldsymbol{\psi}$ (see equation (6.12)) with respect to the coordinates of the reference point of the i-th joint, which only contribute to the corresponding single mask $\boldsymbol{\psi}^i$, are:

$$\frac{\partial \boldsymbol{\psi}}{\partial x^i} = \frac{\partial \boldsymbol{\psi}^i}{\partial x^i} \prod_{j \neq i} \boldsymbol{\psi}^j \tag{E.2a}$$

$$\frac{\partial \boldsymbol{\psi}}{\partial y^i} = \frac{\partial \boldsymbol{\psi}^i}{\partial y^i} \prod_{j \neq i} \boldsymbol{\psi}^j \tag{E.2b}$$

E.3 NDS derivatives for the introduction of holes

To introduce holes in the structure, equation (6.14) is used. The sensitivities with respect to the projected densities and the hole's center location at the reference point of joint i are:

$$\frac{\partial \hat{\varrho}_i}{\partial \bar{\varrho}_i} = {}^{-}\psi_i \tag{E.3a}$$

$$\frac{\partial \hat{\underline{\boldsymbol{\varrho}}}}{\partial x^i} = \bar{\underline{\boldsymbol{\varrho}}} \circ \frac{\partial {}^{-}\boldsymbol{\psi}}{\partial x^i} = \bar{\underline{\boldsymbol{\varrho}}} \circ \frac{\partial {}^{-}\boldsymbol{\psi}^i}{\partial x^i} \prod_{j \neq i} {}^{-}\boldsymbol{\psi}^j \tag{E.3b}$$

$$\frac{\partial \hat{\underline{\boldsymbol{\varrho}}}}{\partial y^i} = \bar{\underline{\boldsymbol{\varrho}}} \circ \frac{\partial {}^{-}\boldsymbol{\psi}}{\partial y^i} = \bar{\underline{\boldsymbol{\varrho}}} \circ \frac{\partial {}^{-}\boldsymbol{\psi}^i}{\partial y^i} \prod_{j \neq i} {}^{-}\boldsymbol{\psi}^j \tag{E.3c}$$

E.4 NDS derivatives for the introduction of material

To add circular or cylindrical features of material to a structure, equation (6.17d) is used. The derivatives are in this case:

$$\frac{\partial \hat{\varrho}_i}{\partial \bar{\varrho}_i} = {}^{+}\psi_i \tag{E.4a}$$

$$\frac{\partial \hat{\boldsymbol{\varrho}}}{\partial x^i} = (\bar{\boldsymbol{\varrho}} - \boldsymbol{I}) \circ \frac{\partial^{+}\boldsymbol{\psi}}{\partial x^i} = (\bar{\boldsymbol{\varrho}} - \boldsymbol{I}) \circ \frac{\partial^{+}\boldsymbol{\psi}^i}{\partial x^i} \prod_{j \neq i} {}^{+}\boldsymbol{\psi}^j \tag{E.4b}$$

$$\frac{\partial \hat{\boldsymbol{\varrho}}}{\partial y^i} = (\bar{\boldsymbol{\varrho}} - \boldsymbol{I}) \circ \frac{\partial^{+}\boldsymbol{\psi}}{\partial y^i} = (\bar{\boldsymbol{\varrho}} - \boldsymbol{I}) \circ \frac{\partial^{+}\boldsymbol{\psi}^i}{\partial y^i} \prod_{j \neq i} {}^{+}\boldsymbol{\psi}^j \tag{E.4c}$$

E.5 NDS derivatives for the introduction of material rings

Ring shaped feature can be projected by equation (6.18) onto a structure. The derivatives are:

$$\frac{\partial \hat{\varrho}_i}{\partial \bar{\varrho}_i} = {}^{+}\psi_i \, {}^{-}\psi_i \tag{E.5a}$$

$$\frac{\partial \hat{\boldsymbol{\varrho}}}{\partial x^i} = \frac{\partial^{-}\boldsymbol{\psi}}{\partial x^i} + (\bar{\boldsymbol{\varrho}} - \boldsymbol{I}) \circ \left(\frac{\partial^{+}\boldsymbol{\psi}}{\partial x^i} \circ {}^{-}\boldsymbol{\psi} + {}^{+}\boldsymbol{\psi} \circ \frac{\partial^{-}\boldsymbol{\psi}}{\partial x^i} \right) \tag{E.5b}$$

$$\frac{\partial \hat{\boldsymbol{\varrho}}}{\partial y^i} = \frac{\partial^{-}\boldsymbol{\psi}}{\partial y^i} + (\bar{\boldsymbol{\varrho}} - \boldsymbol{I}) \circ \left(\frac{\partial^{+}\boldsymbol{\psi}}{\partial y^i} \circ {}^{-}\boldsymbol{\psi} + {}^{+}\boldsymbol{\psi} \circ \frac{\partial^{-}\boldsymbol{\psi}}{\partial y^i} \right) \tag{E.5c}$$

E.6 Derivatives of the volume constraints

The derivatives of the part-wise volume constraint from equation (6.33) with respect to the material and position design variables are:

$$\frac{dh_k}{d\varrho_i} = \sum_j \frac{\partial h_k}{\partial \hat{\varrho}_j} \frac{\partial \hat{\varrho}_j}{\partial \bar{\varrho}_j} \frac{\partial \bar{\varrho}_j}{\partial \tilde{\varrho}_j} \frac{\partial \tilde{\varrho}_j}{\partial \varrho_i} \tag{E.6a}$$

$$\frac{dh_k}{dx_i} = \sum_j \frac{\partial h_k}{\partial \hat{\varrho}_j} \frac{\partial \hat{\varrho}_j}{\partial x_i} \tag{E.6b}$$

with:

$$\frac{\partial h_k}{\partial \hat{\varrho}_j} = \begin{cases} 0, & j \notin \mathbb{L}_k \\ \dfrac{v_j}{\sum_{i \in \mathbb{L}_k} v_i}, & j \in \mathbb{L}_k \end{cases} \tag{E.6c}$$

E.7 Derivatives of the minimum distance constraint

The gradient of the aggregated minimum distance constraint is obtained by direct differentiation of equation (6.36):

$$\frac{dh_{agg}}{dx_k} = -\frac{1}{2} \left(\sum_i \sum_{j<i} \left(s^{ij} + \epsilon \right)^{-p} \right)^{-\frac{1}{2p}-1} \left(\sum_i \sum_{j<i} \left(s^{ij} + \epsilon \right)^{-p-1} \frac{ds^{ij}}{dx_k} \right), \tag{E.7a}$$

$$\frac{ds^{ij}}{dx^i} = -2(x^j - x^i), \qquad \frac{ds^{ij}}{dy^i} = -2(y^j - y^i), \tag{E.7b}$$

$$\frac{ds^{ij}}{dx^j} = 2(x^j - x^i), \qquad \frac{ds^{ij}}{dy^j} = 2(y^j - y^i) \tag{E.7c}$$

E.8 Derivatives of the maximum force constraint

The maximum force constraint for the i-th joint is given in equation (6.41):

$$h_i = \frac{\sqrt{F_x^{i\,2} + F_y^{i\,2} + F_z^{i\,2}}}{\bar{F}_t} - 1 \leq 0 \tag{E.8}$$

The gradient of h_i can be obtained by adjoint sensitivity analysis. For this, the pseudo load vector $\boldsymbol{p}^i$ for the adjoint system is needed (the sum over "dim" means iterating over x, y, and z components):

$$p_j^i = \frac{\partial h_i}{\partial \tilde{u}_j} = \sum_{dim} \frac{\partial h_i}{\partial F_{dim}^i} \frac{\partial F_{dim}^i}{\tilde{u}_j} = \sum_{dim} \frac{\partial h_i}{\partial F_{dim}} l_{dim,j}^i \tag{E.9a}$$

$$\boldsymbol{p}^i = \sum_{dim} \frac{\partial h_i}{\partial F_{dim}} \boldsymbol{l}_{dim}^i \tag{E.9b}$$

The pseudo load vector is thus a linear combination of the selection vectors $\boldsymbol{l}_x^i, \boldsymbol{l}_y^i$, and $\boldsymbol{l}_z^i$ introduced in equation (6.40). The adjoint system to be solved in order to obtain the gradients of the force constraint function for a single joint i thus states (see Appendix D.2):

$$\tilde{\boldsymbol{K}}\boldsymbol{\lambda} = -\boldsymbol{p}^i \tag{E.10}$$

If constraint aggregation via the p-norm aggregation function A_{pn} is performed, the aggregated maximum force constraint h_{agg} from equation (6.42) is obtained:

$$h_{agg} = A_{pn}\left(\left[\frac{\sqrt{\boldsymbol{F}^{i\,T}\boldsymbol{F}^i}}{\bar{F}_t}\right]_{i=1,\dots,n_j}\right) - 1 \leq 0 \tag{E.11}$$

Using the A_{pn} aggregation function, the weights w_i to compose a combined gradient from individual gradients of the input terms are given in equation (3.8). Furthermore, if a combined pseudo load vector $\boldsymbol{p}$ for the adjoint sensitivity analysis to obtain the gradient of h_{agg} is composed, the same weights w_i apply, since the individual vectors $\boldsymbol{p}^i$ are derivatives of the aggregated single terms themselves (cf. equation (E.9a)). The combined pseudo load vector $\boldsymbol{p}$ can thus be assembled in the following way:

$$\boldsymbol{p} = \sum_{i=1}^{n_j} w_i \boldsymbol{p}^i \tag{E.12a}$$

$$= \sum_{i=1}^{n_j} \left(\sum_{k=1}^{n_j} \left(\frac{\sqrt{\boldsymbol{F}^{k\,T}\boldsymbol{F}^k}}{\bar{F}_t}\right)^p\right)^{\frac{1}{p}-1} \left(\frac{\sqrt{\boldsymbol{F}^{i\,T}\boldsymbol{F}^i}}{\bar{F}_t}\right)^{p-1} \boldsymbol{p}^i \tag{E.12b}$$

With the combined pseudo load vector from the above equation, only one adjoint system has to be solved concerning the aggregated maximum force constraint:

$$\tilde{\boldsymbol{K}}\boldsymbol{\lambda} = -\boldsymbol{p} \tag{E.13}$$

The total derivatives of h_{agg} with respect to the material DVs and the joint location DVs are then recovered using the solution $\boldsymbol{\lambda}$ from equation E.10, following the same calculation steps as given in equations (6.27) – (6.31):

$$\frac{dh_{agg}}{d\varrho_i} = \sum_j \frac{\partial h_{agg}}{\partial \hat{\varrho}_j}\frac{\partial \hat{\varrho}_j}{\partial \bar{\varrho}_j}\frac{\partial \bar{\varrho}_j}{\partial \tilde{\varrho}_j}\frac{\partial \tilde{\varrho}_j}{\partial \varrho_i}, \qquad \frac{\partial h_{agg}}{\partial \hat{\varrho}_j} = \boldsymbol{\lambda}^T \frac{\partial \tilde{\boldsymbol{K}}}{\partial \hat{\varrho}_j}\tilde{\boldsymbol{u}}, \qquad \frac{dh_{agg}}{dx_j} = \boldsymbol{\lambda}^T \frac{d\tilde{\boldsymbol{K}}}{dx_j}\tilde{\boldsymbol{u}} \tag{E.14}$$

References

[1] Emmelmann, C., Sander, P., Kranz, J., and Wycisk, E., "Laser Additive Manufacturing and Bionics: Redefining Lightweight Design," *Physics Procedia*, Vol. 12, 2011, pp. 364–368.

[2] Sigmund, O., "On the Optimality of Bone Microstructure," *IUTAM Symposium on Synthesis in Bio Solid Mechanics*, Solid Mechanics and its Applications, Springer, Dordrecht, 1999, pp. 221–234.

[3] Sossou, G., Demoly, F., Montavon, G., and Gomes, S., "An additive manufacturing oriented design approach to mechanical assemblies," *Journal of Computational Design and Engineering*, Vol. 5, No. 1, Jan. 2018, pp. 3–18.

[4] Zhu, J.-H., Zhang, W.-H., and Xia, L., "Topology Optimization in Aircraft and Aerospace Structures Design," *Archives of Computational Methods in Engineering*, Vol. 23, No. 4, Dec. 2016, pp. 595–622.

[5] Kumar, L. J. and Krishnadas Nair, C. G., "Current Trends of Additive Manufacturing in the Aerospace Industry," *Advances in 3D Printing & Additive Manufacturing Technologies*, edited by D. I. Wimpenny, P. M. Pandey, and L. J. Kumar, Springer Singapore, Singapore, 2017, pp. 39–54.

[6] Emmelmann, C., Herzog, D., and Kranz, J., "Design for laser additive manufacturing," *Laser Additive Manufacturing: Materials, Design, Technologies, and Applications*, edited by M. Brandt, Woodhead Publishing Series in Electronic and Optical Materials, Woodhead Publishing, Duxford, UK, Jan. 2017, pp. 259–279.

[7] Kranz, J., *Methodik und Richtlinien für die Konstruktion von laseradditiv gefertigten Leichtbaustrukturen*, Light Engineering für die Praxis, Springer, Berlin, Heidelberg, 2017.

[8] Liu, J., Gaynor, A. T., Chen, S., Kang, Z., Suresh, K., Takezawa, A., Li, L., Kato, J., Tang, J., Wang, C. C. L., Cheng, L., Liang, X., and To, A. C., "Current and future trends in topology optimization for additive manufacturing," *Structural and Multidisciplinary Optimization*, Vol. 57, No. 6, June 2018, pp. 2457–2483.

[9] Wang, F., Christiansen, R. E., Yu, Y., Mørk, J., and Sigmund, O., "Maximizing the quality factor to mode volume ratio for ultra-small photonic crystal cavities," *Applied Physics Letters*, Vol. 113, No. 24, Dec. 2018.

[10] Baandrup, M., Sigmund, O., Polk, H., and Aage, N., "Closing the gap towards super-long suspension bridges using computational morphogenesis," *Nature Communications*, Vol. 11, No. 1, June 2020.

[11] Kriegesmann, B. and Lüdeker, J. K., "Robust compliance topology optimization using the first-order second-moment method," *Structural and Multidisciplinary Optimization*, Vol. 60, No. 1, July 2019, pp. 269–286.

[12] Braga, D. F. O., Tavares, S. M. O., da Silva, L. F. M., Moreira, P. M. G. P., and de Castro, P. M. S. T., "Advanced design for lightweight structures: Review and prospects," *Progress in Aerospace Sciences*, Vol. 69, Aug. 2014, pp. 29–39.

[13] Niu, M., *Airframe structural design*, Conmilit Press Ltd., Hongkong, 1988.

[14] EASA, "Certification Specifications for Large Aeroplanes," Tech. Rep. CS-25, Amendment 12, European Union Aviation Safety Agency, 2012.

[15] Baumgartner, A., Harzheim, L., and Mattheck, C., "SKO (soft kill option): the biological way to find an optimum structure topology," *International Journal of Fatigue*, Vol. 14, No. 6, Nov. 1992, pp. 387–393.

[16] Xie, Y. M. and Steven, G. P., "A simple evolutionary procedure for structural optimization," *Computers & Structures*, Vol. 49, No. 5, Dec. 1993, pp. 885–896.

[17] Querin, O., Steven, G., and Xie, Y., "Evolutionary structural optimisation (ESO) using a bidirectional algorithm," *Engineering Computations*, Vol. 15, No. 8, Jan. 1998, pp. 1031–1048.

[18] Yang, X. Y., Xie, Y. M., Steven, G. P., and Querin, O. M., "Bidirectional Evolutionary Method for Stiffness Optimization," *AIAA Journal*, Vol. 37, No. 11, 1999, pp. 1483–1488.

[19] Aage, N., Andreassen, E., Lazarov, B. S., and Sigmund, O., "Giga-voxel computational morphogenesis for structural design," *Nature*, Vol. 550, No. 7674, Oct. 2017, pp. 84–86.

[20] Liu, H., Hu, Y., Zhu, B., Matusik, W., and Sifakis, E., "Narrow-band topology optimization on a sparsely populated grid," *ACM Transactions on Graphics*, Vol. 37, No. 6, Dec. 2018, pp. 251:1–251:14.

[21] Träff, E. A., Sigmund, O., and Aage, N., "Topology optimization of ultra high resolution shell structures," *Thin-Walled Structures*, Vol. 160, March 2021.

[22] Bendsøe, M. P. and Kikuchi, N., "Generating optimal topologies in structural design using a homogenization method," *Computer Methods in Applied Mechanics and Engineering*, Vol. 71, No. 2, Nov. 1988, pp. 197–224.

[23] Bendsøe, M. P., "Optimal shape design as a material distribution problem," *Structural optimization*, Vol. 1, No. 4, Dec. 1989, pp. 193–202.

[24] Wang, M. Y., Wang, X., and Guo, D., "A level set method for structural topology optimization," *Computer Methods in Applied Mechanics and Engineering*, Vol. 192, No. 1, Jan. 2003, pp. 227–246.

[25] Allaire, G., Jouve, F., and Toader, A.-M., "Structural optimization using sensitivity analysis and a level-set method," *Journal of Computational Physics*, Vol. 194, No. 1, Feb. 2004, pp. 363–393.

[26] Sigmund, O. and Maute, K., "Topology optimization approaches: A comparative review," *Structural and Multidisciplinary Optimization*, Vol. 48, No. 6, Aug. 2013, pp. 1031–1055.

[27] Sigmund, O. and Petersson, J., "Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima," *Structural optimization*, Vol. 16, No. 1, Aug. 1998, pp. 68–75.

[28] Sigmund, O., *Design of material structures using topology optimization*, Ph.D. thesis, Technical University of Denmark, 1994.

[29] Bourdin, B., "Filters in topology optimization," *International Journal for Numerical Methods in Engineering*, Vol. 50, No. 9, Feb. 2001, pp. 2143–2158.

[30] Bruns, T. E. and Tortorelli, D. A., "Topology optimization of non-linear elastic structures and compliant mechanisms," *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, No. 26, March 2001, pp. 3443–3459.

[31] Wang, F., Lazarov, B. S., and Sigmund, O., "On projection methods, convergence and robust formulations in topology optimization," *Structural and Multidisciplinary Optimization*, Vol. 43, No. 6, June 2011, pp. 767–784.

[32] Sun, P. F., Arora, J. S., and Haug Jr, E. J., "Fail-Safe Optimal Design of Structures," *Engineering Optimization*, Vol. 2, No. 1, Jan. 1976, pp. 43–53.

[33] Lüdeker, J. K. and Kriegesmann, B., "Fail-safe optimization of beam structures," *Journal of Computational Design and Engineering*, Vol. 6, No. 3, July 2019, pp. 260–268.

[34] Stolpe, M., "Fail-safe truss topology optimization," *Structural and Multidisciplinary Optimization*, Vol. 60, No. 4, Oct. 2019, pp. 1605–1618.

[35] Jansen, M., Lombaert, G., Schevenels, M., and Sigmund, O., "Topology optimization of fail-safe structures using a simplified local damage model," *Structural and Multidisciplinary Optimization*, Vol. 49, No. 4, Nov. 2013, pp. 657–666.

[36] Zhou, M. and Fleury, R., "Fail-safe topology optimization," *Structural and Multidisciplinary Optimization*, Vol. 54, No. 5, Nov. 2016, pp. 1225–1243.

[37] Ambrozkiewicz, O. and Kriegesmann, B., "Adaptive strategies for fail-safe topology optimization," *EngOpt 2018 Proceedings of the 6th International Conference on Engineering Optimization*, edited by H. Rodrigues, J. Herskovits, C. Mota Soares, A. Araújo, J. Guedes, J. Folgado, F. Moleiro, and J. F. A. Madeira, Springer International Publishing, Cham, 2019, pp. 200–211.

[38] Hederberg, H. and Thore, C.-J., "Topology optimization for fail-safe designs using moving morphable components as a representation of damage," *Structural and Multidisciplinary Optimization*, Aug. 2021.

[39] Wang, H., Liu, J., Wen, G., and Xie, Y. M., "The robust fail-safe topological designs based on the von Mises stress," *Finite Elements in Analysis and Design*, Vol. 171, April 2020.

[40] Cid Bengoa, C., Baldomir, A., Hernández, S., and Romera, L., "Multi-model reliability-based design optimization of structures considering the intact configuration and several partial collapses," *Structural and Multidisciplinary Optimization*, Vol. 57, No. 3, March 2018, pp. 977–994.

[41] Cid, C., Baldomir, A., and Hernández, S., "Probability-damage approach for fail-safe design optimization (PDFSO)," *Structural and Multidisciplinary Optimization*, Aug. 2020.

[42] Martínez-Frutos, J. and Ortigosa, R., "Risk-averse approach for topology optimization of fail-safe structures using the level-set method," *Computational Mechanics*, July 2021.

[43] Martínez-Frutos, J. and Ortigosa, R., "Robust topology optimization of continuum structures under uncertain partial collapses," *Computers & Structures*, Vol. 257, Dec. 2021.

[44] Wu, J., Aage, N., Westermann, R., and Sigmund, O., "Infill Optimization for Additive Manufacturing – Approaching Bone-like Porous Structures," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 24, No. 2, 2018, pp. 1127 – 1140.

[45] Wu, K., Sigmund, O., and Du, J., "Design of metamaterial mechanisms using robust topology optimization and variable linking scheme," *Structural and Multidisciplinary Optimization*, Feb. 2021.

[46] Schmidt, T. and Emmelmann, C., "Abschätzung des Leichtbaupotentials von Sekundärstrukturelementen," *RTeJournal - Fachforum für Rapid Technologie*, Vol. 2015, No. 1, Aug. 2015.

[47] Chickermane, H. and Gea, H. C., "Design of multi-component structural systems for optimal layout topology and joint locations," *Engineering with Computers*, Vol. 13, No. 4, Dec. 1997, pp. 235–243.

[48] Thomas, S., Li, Q., and Steven, G., "Topology optimization for periodic multi-component structures with stiffness and frequency criteria," *Structural and Multidisciplinary Optimization*, Vol. 61, No. 6, June 2020, pp. 2271–2289.

[49] Huang, X. and Xie, M., *Evolutionary Topology Optimization of Continuum Structures: Methods and Applications*, John Wiley & Sons, March 2010.

[50] Zhu, J., Zhang, W., and Beckers, P., "Integrated layout design of multi-component system," *International Journal for Numerical Methods in Engineering*, Vol. 78, No. 6, May 2009, pp. 631–651.

[51] Zhu, J.-H., Gao, H.-H., Zhang, W.-H., and Zhou, Y., "A Multi-point constraints based integrated layout and topology optimization design of multi-component systems," *Structural and Multidisciplinary Optimization*, Vol. 51, No. 2, Feb. 2015, pp. 397–407.

[52] Rakotondrainibe, L., Allaire, G., and Orval, P., "Topology optimization of connections in mechanical systems," *Structural and Multidisciplinary Optimization*, Vol. 61, No. 6, June 2020, pp. 2253–2269.

[53] Sigmund, O. and Torquato, S., "Design of materials with extreme thermal expansion using a three-phase topology optimization method," *Journal of the Mechanics and Physics of Solids*, Vol. 45, No. 6, June 1997, pp. 1037–1067.

[54] Hvejsel, C. F. and Lund, E., "Material interpolation schemes for unified topology and multi-material optimization," *Structural and Multidisciplinary Optimization*, Vol. 43, No. 6, June 2011, pp. 811–825.

[55] Florea, V., Pamwar, M., Sangha, B., and Kim, I. Y., "Simultaneous single-loop multimaterial and multijoint topology optimization," *International Journal for Numerical Methods in Engineering*, Vol. 121, No. 7, 2020, pp. 1558–1594.

[56] Shah, V., Pamwar, M., Sangha, B., and Kim, I. Y., "Material interface control in multimaterial topology optimization using pseudo-cost domain method," *International Journal for Numerical Methods in Engineering*, Vol. 122, No. 2, 2021, pp. 455–482.

[57] Kriegesmann, B., "Robust design optimization with design-dependent random input variables," *Structural and Multidisciplinary Optimization*, Vol. 61, No. 2, 2020, pp. 661–674.

[58] Ambrozkiewicz, O. and Kriegesmann, B., "Density-based shape optimization for fail-safe design," *Journal of Computational Design and Engineering*, Vol. 7, No. 5, 2020, pp. 615–629.

[59] Ambrozkiewicz, O. and Kriegesmann, B., "Simultaneous topology and fastener layout optimization of assemblies considering joint failure," *International Journal for Numerical Methods in Engineering*, Vol. 122, No. 1, 2021, pp. 294–319.

[60] Sigmund, O., "A 99 line topology optimization code written in Matlab," *Structural and Multidisciplinary Optimization*, Vol. 21, No. 2, 2001, pp. 120–127.

[61] Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B. S., and Sigmund, O., "Efficient topology optimization in MATLAB using 88 lines of code," *Structural and Multidisciplinary Optimization*, Vol. 43, No. 1, Jan. 2011, pp. 1–16.

[62] Ferrari, F. and Sigmund, O., "A new generation 99 line Matlab code for compliance topology optimization and its extension to 3D," *Structural and Multidisciplinary Optimization*, Vol. 62, No. 4, Oct. 2020, pp. 2211–2228.

[63] Bendsøe, M. P. and Sigmund, O., *Topology Optimization Theory, Methods, and Applications*, Springer-Verlag Berlin Heidelberg, 2nd ed., 2004.

[64] Harzheim, L., *Strukturoptimierung: Grundlagen und Anwendungen*, Europa Lehrmittel, Haan-Gruiten, 2nd ed., 2014.

[65] Sigmund, O., "Morphology-based black and white filters for topology optimization," *Structural and Multidisciplinary Optimization*, Vol. 33, No. 4-5, April 2007, pp. 401–424.

[66] Clausen, A. and Andreassen, E., "On filter boundary conditions in topology optimization," *Structural and Multidisciplinary Optimization*, Vol. 56, No. 5, 2017, pp. 1147–1155.

[67] Stolpe, M. and Svanberg, K., "An alternative interpolation scheme for minimum compliance topology optimization," *Structural and Multidisciplinary Optimization*, Vol. 22, No. 2, Sept. 2001, pp. 116–124.

[68] Bendsøe, M. P. and Sigmund, O., "Material interpolation schemes in topology optimization," *Archive of Applied Mechanics*, Vol. 69, No. 9-10, 1999, pp. 635–654.

[69] Guest, J. K., "Imposing maximum length scale in topology optimization," *Structural and Multidisciplinary Optimization*, Vol. 37, No. 5, Feb. 2009, pp. 463–473.

[70] Fernández, E., Yang, K.-k., Koppen, S., Alarcón, P., Bauduin, S., and Duysinx, P., "Imposing minimum and maximum member size, minimum cavity size, and minimum separation distance between solid members in topology optimization," *Computer Methods in Applied Mechanics and Engineering*, Vol. 368, Aug. 2020.

[71] Dou, S., "A projection approach for topology optimization of porous structures through implicit local volume control," *Structural and Multidisciplinary Optimization*, Vol. 62, No. 2, Aug. 2020, pp. 835–850.

[72] Sigmund, O., "On the usefulness of non-gradient approaches in topology optimization," *Structural and Multidisciplinary Optimization*, Vol. 43, No. 5, May 2011, pp. 589–596.

[73] Svanberg, K., "The method of moving asymptotes - a new method for structural optimization," *International Journal for Numerical Methods in Engineering*, Vol. 24, No. 2, Feb. 1987, pp. 359–373.

[74] Svanberg, K., "MMA and GCMMA – two methods for nonlinear optimization," Tech. rep., Optimization and Systems Theory, KTH, Stockholm, Sweden, 2007.

[75] Sigmund, O., "Manufacturing tolerant topology optimization," *Acta Mechanica Sinica*, Vol. 25, No. 2, April 2009, pp. 227–239.

[76] Guest, J. K., Prévost, J. H., and Belytschko, T., "Achieving minimum length scale in topology optimization using nodal design variables and projection functions," *International Journal for Numerical Methods in Engineering*, Vol. 61, No. 2, 2004, pp. 238–254.

[77] Park, G.-J., Lee, T.-H., Lee, K. H., and Hwang, K.-H., "Robust Design: An Overview," *AIAA Journal*, Vol. 44, No. 1, 2006, pp. 181–191.

[78] Aage, N., Giele, R., and Andreasen, C. S., "Length scale control for high-resolution three-dimensional level set–based topology optimization," *Structural and Multidisciplinary Optimization*, April 2021.

[79] Christiansen, R. E., Lazarov, B. S., Jensen, J. S., and Sigmund, O., "Creating geometrically robust designs for highly sensitive problems using topology optimization," *Structural and Multidisciplinary Optimization*, Vol. 52, No. 4, Oct. 2015, pp. 737–754.

[80] Schmidt, M.-P., Pedersen, C. B. W., and Gout, C., "On structural topology optimization using graded porosity control," *Structural and Multidisciplinary Optimization*, April 2019.

[81] Zhou, M., Lazarov, B. S., Wang, F., and Sigmund, O., "Minimum length scale in topology optimization by geometric constraints," *Computer Methods in Applied Mechanics and Engineering*, Vol. 293, Aug. 2015, pp. 266–282.

[82] Zhang, W., Yuan, J., Zhang, J., and Guo, X., "A new topology optimization approach based on Moving Morphable Components (MMC) and the ersatz material model," *Structural and Multidisciplinary Optimization*, Vol. 53, No. 6, June 2016, pp. 1243–1260.

[83] Niu, B. and Wadbro, E., "On equal-width length-scale control in topology optimization," *Structural and Multidisciplinary Optimization*, Vol. 59, No. 4, April 2019, pp. 1321–1334.

[84] Lazarov, B. S. and Wang, F., "Maximum length scale in density based topology optimization," *Computer Methods in Applied Mechanics and Engineering*, Vol. 318, May 2017, pp. 826–844.

[85] Fernández, E., Collet, M., Alarcón, P., Bauduin, S., and Duysinx, P., "An aggregation strategy of maximum size constraints in density-based topology optimization," *Structural and Multidisciplinary Optimization*, Vol. 60, No. 5, Nov. 2019, pp. 2113–2130.

[86] Schevenels, M., Lazarov, B. S., and Sigmund, O., "Robust topology optimization accounting for spatially varying manufacturing errors," *Computer Methods in Applied Mechanics and Engineering*, Vol. 200, No. 49–52, 2011, pp. 3613–3627.

[87] Qian, X. and Sigmund, O., "Topological design of electromechanical actuators with robustness toward over- and under-etching," *Computer Methods in Applied Mechanics and Engineering*, Vol. 253, Jan. 2013, pp. 237–251.

[88] Sigmund, O., "On the Design of Compliant Mechanisms Using Topology Optimization," *Mechanics of Structures and Machines*, Vol. 25, No. 4, Jan. 1997, pp. 493–524.

[89] Kreisselmeier, G. and Steinhauser, R., "Systematic Control Design by Optimizing a Vector Performance Index," *IFAC Proceedings Volumes*, Vol. 12, No. 7, Sept. 1979, pp. 113–117.

[90] Kreisselmeier, G. and Steinhauser, R., "Application of vector performance optimization to a robust control loop design for a fighter aircraft," *International Journal of Control*, Vol. 37, No. 2, Feb. 1983, pp. 251–284.

[91] Bertsekas, D. P., *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, NY, USA, 1982.

[92] Qiu, G. Y. and Li, X. S., "A note on the derivation of global stress constraints," *Structural and Multidisciplinary Optimization*, Vol. 40, No. 1, May 2009, pp. 625.

[93] Kennedy, G. J. and Hicken, J. E., "Improved constraint-aggregation methods," *Computer Methods in Applied Mechanics and Engineering*, Vol. 289, June 2015, pp. 332–354.

[94] Raspanti, C. G., Bandoni, J. A., and Biegler, L. T., "New strategies for flexibility analysis and design under uncertainty," *Computers & Chemical Engineering*, Vol. 24, No. 9, Oct. 2000, pp. 2193–2209.

[95] Le, C., Norato, J., Bruns, T., Ha, C., and Tortorelli, D., "Stress-based topology optimization for continua," *Structural and Multidisciplinary Optimization*, Vol. 41, No. 4, 2009, pp. 605–620.

[96] Silva, G. A. d., Aage, N., Beck, A. T., and Sigmund, O., "Three-dimensional manufacturing tolerant topology optimization with hundreds of millions of local stress constraints," *International Journal for Numerical Methods in Engineering*, Vol. 122, No. 2, 2021, pp. 548–578.

[97] Senhora, F. V., Giraldo-Londoño, O., Menezes, I. F. M., and Paulino, G. H., "Topology optimization with local stress constraints: a stress aggregation-free approach," *Structural and Multidisciplinary Optimization*, Vol. 62, No. 4, Oct. 2020, pp. 1639–1668.

[98] Yang, R. J. and Chen, C. J., "Stress-based topology optimization," *Structural Optimization*, Vol. 12, No. 2-3, Oct. 1996, pp. 98–105.

[99] Verbart, A., Langelaar, M., and Keulen, F. v., "A unified aggregation and relaxation approach for stress-constrained topology optimization," *Structural and Multidisciplinary Optimization*, Vol. 55, No. 2, Feb. 2017, pp. 663–679.

[100] Allaire, G., Jouve, F., and Michailidis, G., "Thickness control in structural optimization via a level set method," *Structural and Multidisciplinary Optimization*, Vol. 53, No. 6, June 2016, pp. 1349–1382.

[101] Ambrozkiewicz, O. and Kriegesmann, B., "Fail-safe design of topology optimized structures using density-based member shaping," *5th ECCOMAS Young Investigators Conference*, Krakow, Poland, Sept. 2019.

[102] Rozvany, G. I. N., "A critical review of established methods of structural topology optimization," *Structural and Multidisciplinary Optimization*, Vol. 37, No. 3, Jan. 2009, pp. 217–237.

[103] Bremicker, M., Chirehdast, M., Kikuchi, N., and Papalambros, P. Y., "Integrated Topology and Shape Optimization in Structural Design," *Mechanics of Structures and Machines*, Vol. 19, No. 4, Jan. 1991, pp. 551–587.

[104] Lin, C.-Y. and Chao, L.-S., "Automated image interpretation for integrated topology and shape optimization," *Structural and Multidisciplinary Optimization*, Vol. 20, No. 2, Oct. 2000, pp. 125–137.

[105] Lüdeker, J. and Kriegesmann, B., "Shape optimization of topologically optimized structures using NURBS," *27. DfX-Symposium*, Jesteburg, Germany, Oct. 2016.

[106] Hojjat, M., Stavropoulou, E., and Bletzinger, K.-U., "The Vertex Morphing method for node-based shape optimization," *Computer Methods in Applied Mechanics and Engineering*, Vol. 268, Jan. 2014, pp. 494–513.

[107] Le, C., Bruns, T., and Tortorelli, D., "A gradient-based, parameter-free approach to shape optimization," *Computer Methods in Applied Mechanics and Engineering*, Vol. 200, No. 9, Feb. 2011, pp. 985–996.

[108] Lüdeker, J. K., Sigmund, O., and Kriegesmann, B., "Inverse homogenization using isogeometric shape optimization," *Computer Methods in Applied Mechanics and Engineering*, Vol. 368, Aug. 2020.

[109] Lazarov, B. S., Schevenels, M., and Sigmund, O., "Topology optimization with geometric uncertainties by perturbation techniques," *International Journal for Numerical Methods in Engineering*, Vol. 90, No. 11, 2012, pp. 1321–1336.

[110] Duysinx, P. and Bendsøe, M. P., "Topology optimization of continuum structures with local stress constraints," *International Journal for Numerical Methods in Engineering*, Vol. 43, 1998, pp. 1453–1478.

[111] Amir, O. and Lazarov, B. S., "Achieving stress-constrained topological design via length scale control," *Structural and Multidisciplinary Optimization*, June 2018, pp. 1–19.

[112] Chickermane, H., Gea, H. C., Yang, R. J., and Chuang, C. H., "Optimal fastener pattern design considering bearing loads," *Structural optimization*, Vol. 17, No. 2, April 1999, pp. 140–146.

[113] Barlow, J., "Constraint relationships in linear and nonlinear finite element analyses," *International Journal for Numerical Methods in Engineering*, Vol. 18, No. 4, 1982, pp. 521–533.

[114] Torii, A. J. and Faria, J. R. d., "Structural optimization considering smallest magnitude eigenvalues: a smooth approximation," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, Vol. 39, No. 5, May 2017, pp. 1745–1754.

[115] Haldar, A. and Mahadevan, S., *Probability, Reliability and Statistical Methods in Engineering Design*, John Wiley & Sons, New York ; Chichester England, 1st ed., Nov. 1999.

[116] Gamache, J.-F., Vadean, A., Noirot-Nérin, m., Beaini, D., and Achiche, S., "Image-based truss recognition for density-based topology optimization approach," *Structural and Multidisciplinary Optimization*, June 2018, pp. 1–13.

[117] Weiss, B. M., Hamel, J. M., Ganter, M. A., and Storti, D. W., "Data-Driven Additive Manufacturing Constraints for Topology Optimization," *Journal of Manufacturing Science and Engineering*, Vol. 143, No. 021001, Oct. 2020.

[118] Zhang, W., Zhong, W., and Guo, X., "An explicit length scale control approach in SIMP-based topology optimization," *Computer Methods in Applied Mechanics and Engineering*, Vol. 282, Dec. 2014, pp. 71–86.

[119] Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 1, Jan. 1979, pp. 62–66.

[120] Yetis, F. A. and Saitou, K., "Decomposition-Based Assembly Synthesis Based on Structural Considerations," *Journal of Mechanical Design*, Vol. 124, No. 4, Nov. 2002, pp. 593–601.

[121] Yamada, T., "Geometric shape features extraction using a steady state partial differential equation system," *Journal of Computational Design and Engineering*, Vol. 6, No. 4, Oct. 2019, pp. 647–656.

[122] Pratt, W. K., *Digital Image Processing*, Wiley, June 2006.

[123] Dou, S. and Stolpe, M., "Fail-safe optimization of tubular frame structures under stress and eigenfrequency requirements," *Computers & Structures*, Vol. 258, Jan. 2022.

[124] Dou, S. and Stolpe, M., "On stress-constrained fail-safe structural optimization considering partial damage," *Structural and Multidisciplinary Optimization*, Jan. 2021.

[125] Kumar, P. and Fernández, E., "A numerical scheme for filter boundary conditions in topology optimization on regular and irregular meshes," *arXiv:2101.01122 [cs, math]*, Jan. 2021.

[126] Díaz, A. and Sigmund, O., "Checkerboard patterns in layout optimization," *Structural optimization*, Vol. 10, No. 1, Aug. 1995, pp. 40–45.

[127] Bathe, K.-J., *Finite Element Procedures*, Klaus-Jürgen Bathe, S.l., second edition ed., 2014.

[128] Haug, E. J., Choi, K. K., and Komkov, V., *Design Sensitivity Analysis of Structural Systems*, Academic Press, Orlando, Florida, USA, 1986.