Journal of Computational Design and Engineering, 2020, 7(0), 1–15

doi: 10.1093/jcde/qwaa044
Journal homepage: www.jcde.org

RESEARCH ARTICLE

Density-based shape optimization for fail-safe design

Olaf Ambrozkiewicz * and Benedikt Kriegesmann

Hamburg University of Technology, Am Schwarzenberg-Campus 4, Working group structural optimization for lightweight design, 21073 Hamburg, Germany

*Corresponding author. E-mail: olaf.ambrozkiewicz@tuhh.de http://orcid.org/0000-0003-3250-7095

Abstract

This paper presents a two-stage procedure for density-based optimization towards a fail-safe design. Existing approaches either are computationally extremely expensive or do not explicitly consider fail-safe requirements in the optimization. The current approach trades off both aspects by employing two sequential optimizations to deliver redundant designs that offer robustness to partial failure. In the first stage, a common topology optimization or 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 current approach are demonstrated by application to various 2D and 3D examples. The results show that, in contrast to explicitly enforcing fail-safety in topology optimization, the current approach can be carried out with reasonable computational cost. Compared to the local volume constraint approach, the suggested procedure further increases the fail-safe performance by 47% for the example considered.

Keywords: fail-safe; topology optimization; shape optimization

1. Introduction

Safety-critical structures often need to be designed such that they resist a certain load level, even after one load path failed. For instance, according to the Certification Specifications for Large Aeroplanes CS-25 by the European Aviation Safety Agency (2012), a "multiple load path construction" is mandatory 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 the design load, even if one load path fails. This ability is also often referred to as damage tolerance in the aircraft industry. Motivated by the advances in additive manufacturing, according to Sossou, Demoly, Montavon, and Gomes (2018), design optimization and in particular topology optimization is increasingly used for the design of aircraft structures. Therefore, it is desirable to consider fail-safe requirements already at the optimization stage.

Embedding the fail-safe requirements in structural optimization is straight forward and has already been done by Sun, Arora, and Haug (1976), where the fail-safe design in the optimization 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.

Implementing fail-safe requirements in topology optimization raises the question of how to remove structural elements, when these evolve during the optimization. Jansen, Lombaert, Schevenels, and Sigmund (2013) tackled this problem by introducing a simplified damage model using damage patches of fixed shape. Since the location of damage is not known a priori, a vast number of possible damage locations have to be evaluated in each iteration. The result is a structure with redundant load paths, but the computational cost is extremely high.

Received: 14 October 2019; **Revised:** 20 March 2020; **Accepted:** 6 February 2020

© The Author(s) 2020. Published by Oxford University Press on behalf of the Society for Computational Design and Engineering. This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact journals.permissions@oup.com

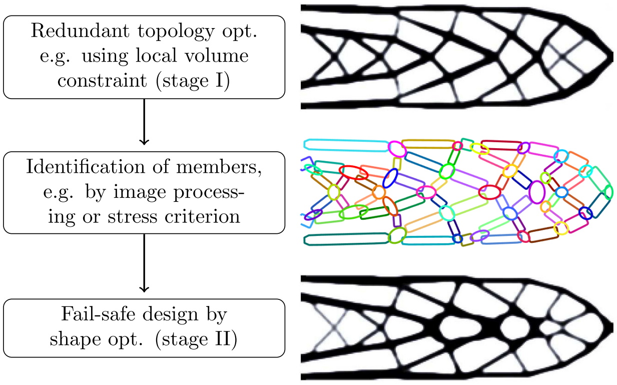


Figure 1: Workflow for proposed fail-safe optimization.

Zhou and Fleury (2016) presented an approach to downselect the locations at which damage patches are to be placed. This reduced the computational cost, that however remains on a very high level. Furthermore, their approach produces designs that are sensitive to damage in locations that have been sorted out (see Ambrozkiewicz and Kriegesmann 2018). Lüdeker and Kriegesmann (2019) discussed approaches for combining failure cases in order to improve computational efficiency. They showed that this is not necessarily a conservative approach, and moreover, the number of analyses to be run per iteration remains unacceptably high.

The authors Ambrozkiewicz and Kriegesmann (2018) presented a method for fail-safe optimization where structural elements are identified as “beams” and “knots” during the optimization. Based on this identification, the damage patch size, shape, and location are adaptively adjusted during the optimization to cover all load-bearing parts of the structure, preferably in a minimal amount of damage patches. In the authors’ opinion, this approach is close to the intention of the Certification Specifications for Large Aeroplanes. The number of damage patches to be considered per iteration decreases by orders of magnitude when compared to the method in Jansen et al. (2013). However, since the optimization problem is continuously changed, the optimization does not converge reliably in all cases.

Besides explicitly formulating the fail-safe requirement as objective or constraint in the optimization, a redundant structure can also be achieved by using the local volume constraints introduced by Wu, Aage, Westermann, and Sigmund (2017). 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.

The current paper aims at combining the advantages of the approach suggested in Wu et al. (2017) with an explicit formulation of the fail-safe requirement as in Jansen et al. (2013) by a sequential optimization procedure. The workflow for the proposed fail-safe optimization is shown in Fig. 1.

The first step consists of carrying out a topology optimization with the local volume constraint method. Afterward, the structural members are identified. Finally, a density-based shape optimization is carried out in which only the failure of the previously identified structural elements is considered. Thereby, the huge efficiency improvement as in Ambrozkiewicz and Kriegesmann (2018) is driven even further. By limiting the second optimization step to a shape optimization, the convergence behavior is not jeopardized.

This paper outlines as follows: In Section 2, an overview of the proposed method is given, while later sections describe the details of the individual calculation steps. Section 3 recapitulates the theory of topology optimization. The identification of load-bearing parts and the modeling of damage scenarios are explained in Section 4. Density-based shape optimization with fail-safe consideration is the main contribution of this paper and described in Section 5. Numerical examples in 2D are given in Section 6, 3D examples can be found in Section 7. A discussion of computational cost in Section 8 sets the proposed method in comparison to the methods of Jansen et al. (2013), and Zhou and Fleury (2016).

2. Sequential Topology and Shape Optimization for Fail-Safe Design

As shown in Fig. 1, the proposed method consists of these main steps:

(1) Deterministic topology optimization to get a preliminary design I
(2) Identification of load-bearing members of I and creation of damage scenarios
(3) Density-based shape optimization of design I to get a fail-safe design II

In the first step, a computationally cheap topology optimization of the part is run without any fail-safe considerations (see Section 3). This stage will define the final topology of the part. Since redundant structures are most likely more robust to partial damage, using the local volume constraint (see Section 3.7) is a good choice if aiming for optimized damage tolerance.

Afterward, the load-bearing parts of the structure are identified by an automated algorithm (see Section 4) and damage scenarios are generated.

To finally evaluate and minimize the effect of damage on these load paths is the task of the density-based shape optimization stage (see Section 5).

3. Topology Optimization of Redundant Structures – Stage I

Minimizing compliance c under a volume constraint h is a well-studied topology optimization task that can be performed to deliver the preliminary design I of the structure.

This section recapitulates the equations for deterministic topology optimization problems under a global volume constraint. Later, the local volume constraint is presented as a cheap modification of the problem formulation to obtain redundant structures.

3.1. Minimum compliance optimization formulation

The optimization task states as follows:

$$\min_{\varrho} c(\varrho) = \boldsymbol{f}^{T}\boldsymbol{u} \tag{1}$$

$$\text{s.t.} \quad h(\varrho) = V(\varrho)/V_0 - k_g \leq 0 \tag{2}$$

$$0 \leq \varrho_i \leq 1, \forall i \tag{3}$$

$$K(\varrho)\boldsymbol{u} = \boldsymbol{f} \tag{4}$$

The stiffness matrix $\boldsymbol{K}$ is dependent on the vector of design variables $\boldsymbol{\varrho}$ and is obtained by a finite element discretization of the design domain. The nodal displacement vector $\boldsymbol{u}$ is the solution to the FEM equation of linear elasticity (4) under an external load vector $\boldsymbol{f}$. V_0 is the total volume of the design domain and V is the volume occupied by the part which is limited by an upper bound k_g on the global volume fraction V/V_0.

3.2. Filtering of the design variables

The design variables $\boldsymbol{\varrho}$ are mapped to filtered variables $\tilde{\boldsymbol{\varrho}}$ by a mesh independency filter (Bourdin 2001; Bruns and Tortorelli 2001):

$$\tilde{\varrho}_i = \frac{\sum_j w(\mathbf{x}_j - \mathbf{x}_i) v_j \varrho_j}{\sum_j w(\mathbf{x}_j - \mathbf{x}_i) v_j} \tag{5}$$

$$w(\mathbf{x}) := \max(r - \|\mathbf{x}\|_2, 0) \tag{6}$$

Here, v_i are the element volumes, $\mathbf{x}_i$ the locations of the element centers and w is a conic weighting function dependent on the filter radius r.

3.3. Projection of the filtered variables

The filtered variables $\tilde{\boldsymbol{\varrho}}$ are then transformed to projected variables $\bar{\boldsymbol{\varrho}}$ via an approximation to the Heaviside step-function with a variable steepness parameter β and the threshold parameter η (Wang, Lazarov, and Sigmund 2011):

$$\bar{\varrho}_i = \frac{\tanh(\beta\eta) + \tanh(\beta(\tilde{\varrho}_i - \eta))}{\tanh(\beta\eta) + \tanh(\beta(1 - \eta))} \tag{7}$$

3.4. Material interpolation

Finally, the element's stiffness matrix for element i is assembled based on an effective Young's modulus E_i calculated by the SIMP approach dependent on the projected element variable $\bar{\varrho}_i$ (Bendsøe 1989):

$$E_i = E_{min} + (E_0 - E_{min})(\bar{\varrho}_i)^p, \qquad \bar{\varrho}_i \in [0, 1] \tag{8}$$

For the material interpolation, E_0 and E_{min} are set as the Young's moduli of the solid and void phase, respectively. The exponent $p > 1$ is used to penalize intermediate densities.

3.5. Global volume constraint

With the pseudo-densities $\bar{\varrho}_i$ and the element volumes v_i the global volume fraction ω can be defined, and the constraint equation (2) can be expressed as

$$\omega = \frac{V}{V_0} = \frac{\sum_i \bar{\varrho}_i v_i}{\sum_i v_i} \tag{9}$$

$$h = \omega - k_g \leq 0 \tag{10}$$

3.6. Sensitivities

When using gradient-based optimization algorithms, the sensitivities of the objective function c and the constraint equation h are required. Firstly, the partial derivatives with respect to the pseudo-densities are calculated. For c the adjoint method is used (Bendsøe and Sigmund 2004), whereas for h direct differentiation of equations (9) and (10) is performed:

$$\frac{\partial c}{\partial \bar{\varrho}_i} = -\boldsymbol{u}^T \frac{d\boldsymbol{K}}{d\bar{\varrho}_i} \boldsymbol{u} \tag{11}$$

$$\frac{\partial h}{\partial \bar{\varrho}_i} = \frac{v_i}{\sum_j v_j} \tag{12}$$

Then, the total derivatives with respect to the design variables are obtained by applying the chain rule:

$$\frac{dc}{d\varrho_i} = \cancelto{0}{\frac{\partial c}{\partial \varrho_i}} + \sum_j \frac{\partial c}{\partial \bar{\varrho}_j} \frac{\partial \bar{\varrho}_j}{\partial \tilde{\varrho}_j} \frac{\partial \tilde{\varrho}_j}{\partial \varrho_i} \tag{13}$$

$$\frac{dh}{d\varrho_i} = \cancelto{0}{\frac{\partial h}{\partial \varrho_i}} + \sum_j \frac{\partial h}{\partial \bar{\varrho}_j} \frac{\partial \bar{\varrho}_j}{\partial \tilde{\varrho}_j} \frac{\partial \tilde{\varrho}_j}{\partial \varrho_i} \tag{14}$$

3.7. Local volume constraint

A cheap way to get redundant load paths in a topology optimization is to suppress the forming of overly thick members by constraining volume fractions locally (Wu et al. 2017). For the local volume constraint, the volume fraction ω_i in a local neighborhood $\mathbb{N}_i$ around each element i is constrained by a maximum allowed value k_l:

$$\omega_i = \frac{\sum_{j \in \mathbb{N}_i} \bar{\varrho}_j v_j}{\sum_{j \in \mathbb{N}_i} v_j} \tag{15}$$

$$h_i = \omega_i - k_l \leq 0 \tag{16}$$

With $\mathbb{N}_i$ being the neighborhood of element i including all surrounding elements, that have their centers within a maximum distance radius of R:

$$\mathbb{N}_i = \{ j \mid \|\mathbf{x}_j - \mathbf{x}_i\|_2 \leq R \} \tag{17}$$

By enforcing a certain amount of void in each neighborhood, thick members need to be split into separate thinner members with enough void space in between to satisfy the local volume constraint. This leads to a redundant design, where the degree of redundancy may be tuned by the relation of the filter radius r and the radius R of the local volume constraints.

The n constraints h_i may be combined into a single constraint h_{agg} by applying the p-mean aggregation function on the local volume fractions ω_i (Wu et al. 2017):

$$h_{agg} = \left(\frac{1}{n} \sum_i \omega_i^{p_{agg}} \right)^{\frac{1}{p_{agg}}} - k_l \leq 0 \tag{18}$$

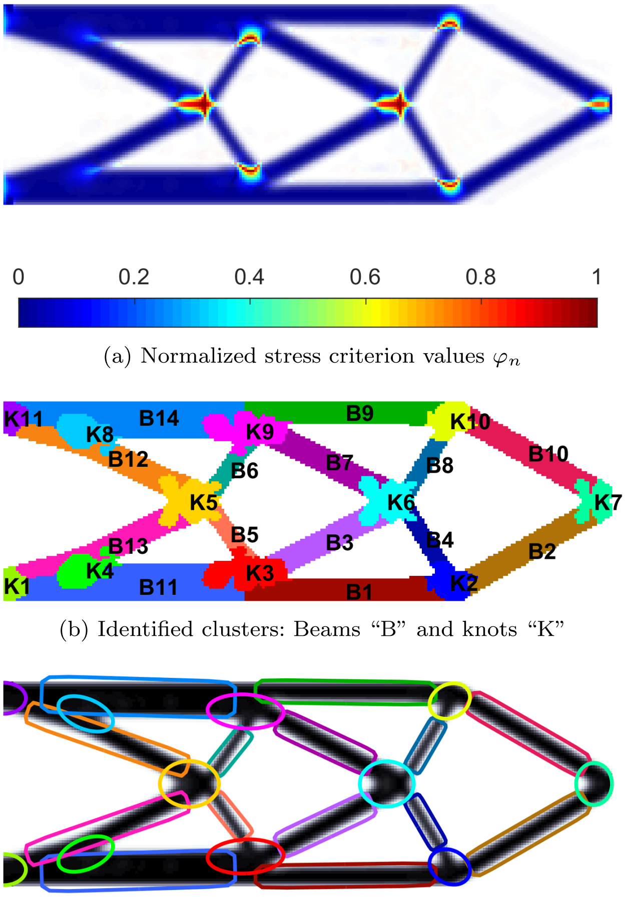
This aggregation approximates the maximum value of all ω_i in a differentiable way. It converges to the real maximum with increasing values for the parameter p_{agg}, with the downside that the nonlinearity of the constraint is increased. In this paper a value of $p_{agg} = 16$ is used (Wu et al. 2017).

The minimum compliance topology optimization task with local volume constraint is obtained by taking equations (1)–(4) and changing the inequality constraint (2) to (18).

4. Modeling Damage Scenarios on Load-Bearing Members

After a design is obtained in optimization stage I, individual damage scenarios have to be defined to tackle the original problem of considering failure of single load-bearing members of the structure. This section describes the proposed method of defining the damage scenarios, consisting of these steps:

(1) Identification of load-bearing members
(2) Analytical description of individual members by auxiliary "member shapes"



(a) Normalized stress criterion values φ_n

(b) Identified clusters: Beams "B" and knots "K"

(c) Auxiliary shapes on density field

Figure 2: Load-path identification for the cantilever beam example.

(3) Conversion of "member shapes" to "damage shapes"
(4) Generation of damage mask vectors $\boldsymbol{\psi}_d$

4.1. Identification of load-bearing members

Load-bearing members are extracted by subdividing the solid phase from the result obtained in stage I into "beams" and "knots" (c.f. Fig. 2b). 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 A describes the details of two methods successfully applied by the authors based on a simple stress criterion (Ambrozkiewicz and Kriegesmann 2018) and image processing algorithms.

Due to the automated methods available, this paper focuses on 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 utilizing the local volume constraint, even for 3D problems lattice-type structures are enforced.

As an example, the stress criterion method by Ambrozkiewicz and Kriegesmann (2018) is applied to the cantilever beam structure from Section 6.1 with the density field seen in Fig. 6a. The normalized stress criterion values for the elements of the solid phase are shown in Fig. 2a. Within the value set $\varphi_n \in [0, 1]$, low values indicate a beam region, higher values are obtained for knot regions.

The classification into beams and knots is done by selecting a threshold value for the criterion, e.g. by choosing a specific probability value and reading the threshold value from the empirical cumulative distribution function of the criterion values. In this example, the structure is dominated by slender beams with relatively little material in the interconnection regions. Consequently, a probability value of 0.7 was chosen, such that 70% of the elements were classified into the beam class and the remaining 30% into the knot class. This procedure is robust towards moderate variations in the exact threshold value, since altering it would only let the knot regions grow or shrink by a small amount, without negatively affecting the overall subdivision process.

After each solid element is assigned to a class (beam or knot), a recursive algorithm collects neighboring elements of the same class. Each individual group is assigned a cluster number. In an automated post-processing step, undesired small clusters are suppressed and too large clusters are split, providing the result in Fig. 2b. Appendix A discusses the classification and post-processing steps in more detail.

4.2. Parametric description of auxiliary member shapes

Auxiliary "member shapes" are constructed for each identified cluster: 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. To generalize the shapes, superellipses (in 2D) and superellipsoids (in 3D) are used to describe the member shapes.

For every cluster, the following geometric quantities are needed to describe the auxiliary shapes mathematically:

- The location of the center point $\boldsymbol{x}_c$
- A matrix $\mathbf{V}$ with unit vectors of the principal axes as columns
- A size vector $\boldsymbol{a}$ with the elongation along the half-axes

Appendix B describes how these quantities are calculated for each of the clusters.

The surface for the outline of a superellipsoid (for the 2D case the z-coordinate can be omitted, leading to a superellipse) in local coordinates (x', y', z') with the center at $(0, 0, 0)$ is

$$E(\boldsymbol{x}') := \left|\frac{x'}{a}\right|^q + \left|\frac{y'}{b}\right|^q + \left|\frac{z'}{c}\right|^q - 1 = 0 \qquad (19)$$

Here $\boldsymbol{a} = [a, b, c]^T$ are the lengths of the half-axes, q is the power to alter the shape of the edge transitions: For the auxiliary shapes for knots a value of $q = 2$ will be used, a higher value of $q = 10$ is used for beams, resulting in almost box-type shapes.

In general, the superellipsoids will have their centers shifted by a position vector to the center point $\boldsymbol{x}_c$ and their principal axes inclined compared to the global coordinate system. The principal axes will be defined as an orthonormal basis $\mathbf{V}$:

$$\mathbf{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3] \qquad (20)$$

The relation between global coordinates $\boldsymbol{x}$ and local coordinates $\boldsymbol{x}'$ is then

$$\boldsymbol{x}' = \mathbf{V}^T(\boldsymbol{x} - \boldsymbol{x}_c) \qquad (21)$$

An overlay of the outlines of the member shapes calculated from the clusters in Fig. 2b on the example design is shown in Fig. 2c. With the scaling of the size vector $\boldsymbol{a}$ derived in Appendix B, a close fit of the outlines of the member shapes to the original density field is achieved.

4.3. Damage shapes from member shapes

Each damage scenario is modeled by removing material from the structure inside geometric entities referred to as “damage shapes” in this paper. Parameterizing the shapes’ sizes and locations means that, in contrast to the approach from Jansen et al. (2013), the geometry of the damage shapes used here is not constant and the shapes are not necessarily aligned with the FE mesh axes.

The previously obtained member shapes cannot be directly used as regions of damage for a downstream fail-safe optimization. The shapes need to be enlarged, otherwise, the optimizer would simply circumvent the damage by slightly shifting the outline of the part. A simple approach to counter this undesired effect is to use upscaled versions of the member shapes as damage shapes e.g. by multiplying the size vector $\boldsymbol{a}$ by 2. The circumvention effect and the influence of the scaling factor on the final optimization result is discussed in Appendix C.

The fail-safe optimization procedure described in Section 5 is not sensitive to the exact size of the damage shapes, they only need to be large enough to reliably cut through the corresponding structural member in the density field. Beam-type members only need to be cut in the middle since beams lose all their load-bearing capacity once they are cut through their whole cross-section at any location. These more compact damage shapes are used for the actual fail-safe optimization.

An example of the final damage shapes can be seen in Fig. 6b, where a scaling factor of 2 was applied to the size vector $\boldsymbol{a}$ obtained for the member shapes from Fig. 2c. For beams, more compact centered damage shapes were obtained by modifying the largest half-axis (length direction) to be half the size of the second half-axis (width direction): $a = 0.5b$.

4.4. Generating damage mask vectors

A multiplicative mask $\boldsymbol{\psi}_d$ is used to describe every damage scenario d. The physical densities of the damaged structure are the Hadamard (entrywise) product of the undamaged projected densities and the damage mask:

$$\bar{\boldsymbol{\varrho}}_d = \bar{\boldsymbol{\varrho}} \circ \boldsymbol{\psi}_d, \forall d \tag{22}$$

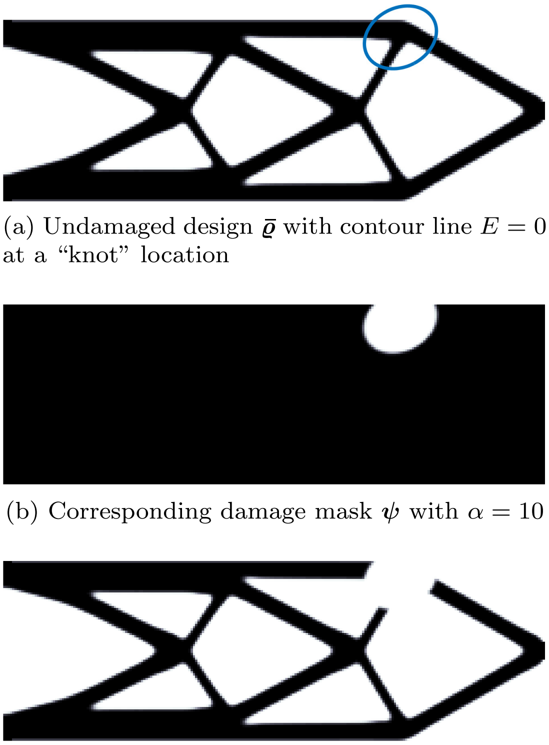
The entries of $\boldsymbol{\psi}_d$ are values in the interval from 0 to 1, with 1 meaning that the structure is unchanged and 0 meaning that material is completely removed in this area. Also, partial removing is possible when using values between 0 and 1.

To generate the mask for a damage case d, all elements with center points inside the corresponding damage shape superellipsoid $(E(\mathbf{x}') \leq 0)$ shall get a mask value close to zero, while all other elements outside $(E(\mathbf{x}') > 0)$ shall be mapped to one. A smooth approximation for this mapping function can be constructed by using the hyperbolic tangent function. For each element i with the center point at $\mathbf{x}_i$ the mask value is

$$\psi_{d,i} = \frac{\tanh(\alpha E(\mathbf{x}_i')) + 1}{2} \tag{23}$$

$$\mathbf{x}_i' = \mathbf{V}^T(\mathbf{x}_i - \mathbf{x}_c) \tag{24}$$

The parameter α controls the sharpness of the transition between zero and one and is set to $\alpha = 10$ in this paper. An example of applying a single damage patch to a structure by the described method is shown in Fig. 3.


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

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

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

Figure 3: Example of applying damage to a structure by a multiplicative mask $\boldsymbol{\psi}$ generated from a parameterization E of an ellipse.

5. Fail-Safe Design by Density-Based Shape Optimization – Stage II

In a shape optimization, 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 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. Hojjat, Stavropoulou, and Bletzinger 2014; Le, Bruns, and Tortorelli 2011).

In contrast to that, the density-based shaping optimization presented here uses the same mesh as for the topology optimization, therefore no remeshing nor data conversion or mapping is needed.

The interface of the structure optimized in stage I using the SIMP approach with relaxed continuous density variables is not sharp but blurred out due to the averaging effect of the density filter (5). The projection step (7) is applied to make the gradual interface region more compact for larger β values such that the interface appears sharper.

However, when modifying the projection threshold value η, the outline of the part can be altered. This was used e.g. to include the effects of uniform (see Sigmund 2009; Wang et al. 2011) or spatially varying (see Kriegesmann and Lüdeker 2019; Lazarov, Schevenels, and Sigmund 2012; Schevenels, Lazarov, and Sigmund 2011) manufacturing errors into the optimization,

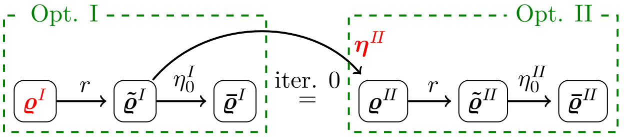


Figure 4: Relation of variables for the two optimization stages, the design variables are displayed in red.

where eroded and dilated designs were calculated besides the nominal design by altering η values. In Appendix D, the effect of uniformly varying η is shown for an example structure.

In this paper 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. Where needed, the superscripts $(\cdot)^{I}$ or $(\cdot)^{II}$ are included to clarify if variables originate from the preliminary design of stage I or from the shape optimization of stage II discussed in this section. Figure 4 shows the relations of the variables and is intended as a help to keep track of which variables are used in the individual optimization stages.

5.1. Design variables of stage II

In stage II the $\boldsymbol{\eta}^{II}$ values take the role of design variables, whereas the final filtered variables $\tilde{\boldsymbol{\varrho}}^{I}$ of the preliminary design I are used as the basis and stay constant (cf. Fig. 4):

$$\boldsymbol{\varrho}^{II} = \boldsymbol{\varrho}^{II}(\tilde{\boldsymbol{\varrho}}^{I}, \boldsymbol{\eta}^{II}, \beta) \tag{25}$$

5.2. Filtering of stage II

Since now with the element-wise η_i^{II} each physical density $\bar{\varrho}_i^{II}$ can again be modified independently of its neighbors, the effect of the density filter of stage I is nullified: Checkerboarding may occur again and mesh independency is lost. Therefore, another filtering and projection step with a constant uniform $\eta_0^{II} = 0.5$ is applied:

$$\tilde{\boldsymbol{\varrho}}^{II} = \tilde{\boldsymbol{\varrho}}^{II}(\boldsymbol{\varrho}^{II}, r) \tag{26}$$

$$\bar{\boldsymbol{\varrho}}^{II} = \bar{\boldsymbol{\varrho}}^{II}(\tilde{\boldsymbol{\varrho}}^{II}, \eta_0^{II}, \beta) \tag{27}$$

Note that the constant scalar η_0^{II} is a different variable than the element-wise defined $\boldsymbol{\eta}^{II}$ which will vary spatially and is the design variable of stage II.

The differences between the two optimization stages may 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 by $\boldsymbol{\eta}^{II}$. $\boldsymbol{\eta}^{II}$ is now the design vector and is altered by the optimizer.

5.3. Volume constraint in stage II

To make the results of the stages I and II comparable, the global volume constraint function h from equation (10) is used to prohibit that the total volume of design II exceeds the volume of design I.

5.4. Min–max problem of all damage scenarios

To optimize for a fail-safe design, the effect of all damage scenarios, each degrading a single load-bearing member, has to be evaluated. A separate FE calculation is performed to calculate the structural response for every damage scenario d.

The individual damaged configurations are obtained by applying the damage mask $\boldsymbol{\psi}_d$ to the projected densities using equation (22). The compliance c_d of the damaged structure is therefore a function of $\boldsymbol{\eta}^{II}$ and $\boldsymbol{\psi}_d$:

$$c_d = c(\boldsymbol{\eta}^{II}, \boldsymbol{\psi}_d), \forall d \tag{28}$$

The compliance of the damage scenario with maximum influence should be minimized in optimization stage II. This leads to a min–max problem:

$$\min_{\boldsymbol{\eta}^{II}} \max_{d} c_d \tag{29}$$

By using the Kreisselmeier–Steinhauser formula (Kreisselmeier and Steinhauser 1983), the min–max problem for the objective function is approximated by a differentiable form with a scaling factor γ. The problem for the stage II optimization with a global volume constraint states now:

$$\min_{\boldsymbol{\eta}^{II}} \frac{1}{\gamma} \log\left(\sum_d \exp\left(\gamma c_d\right)\right) \tag{30}$$

$$\text{s.t.} \qquad h = \omega - k_g \leq 0 \tag{31}$$

$$0 \leq \eta_i^{II} \leq 1, \forall i \tag{32}$$

$$\boldsymbol{K}(\boldsymbol{\eta}^{II})\boldsymbol{u} = \boldsymbol{f} \tag{33}$$

5.5. Sensitivities related to the undamaged configuration

Derivatives with respect to $\boldsymbol{\eta}^{II}$ are required for the shape optimization of stage II. The volume constraint h will be calculated for the undamaged configuration. Compared to the gradient used for topology optimization from equation (14), the chain needs to be augmented by another partial derivative term:

$$\frac{dh}{d\eta_i^{II}} = \sum_j \frac{\partial h}{\partial \bar{\varrho}_j^{II}} \frac{\partial \bar{\varrho}_j^{II}}{\partial \tilde{\varrho}_j^{II}} \frac{\partial \tilde{\varrho}_j^{II}}{\partial \varrho_i^{II}} \frac{\partial \varrho_i^{II}}{\partial \eta_i^{II}} \tag{34}$$

For the additional partial derivative, the projection equation (7) has to be differentiated for the threshold parameter η.

5.6. Sensitivities related to damaged configurations

In optimization stage II, the compliances c_d under damage conditions will be calculated. Applying the adjoint equation (11) only yields the derivative of c_d with respect to the damaged projected densities $\bar{\boldsymbol{\varrho}}_d$.

By incorporating the damage masks $\boldsymbol{\psi}_d$ into the gradients, derivatives obtained with respect to the damaged configurations can be transformed into derivatives with respect to the undamaged configuration:

$$\frac{\partial(\cdot)}{\partial \bar{\varrho}_j} = \frac{\partial(\cdot)}{\partial \bar{\varrho}_{d,j}} \frac{\partial \bar{\varrho}_{d,j}}{\partial \bar{\varrho}_j} = \frac{\partial(\cdot)}{\partial \bar{\varrho}_{d,j}} \psi_{d,j} \tag{35}$$

If material for an element j is completely removed by a damage scenario ($\psi_{d,j} = 0$), 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.

Augmenting the gradient (13) with the procedure described in Section 5.5 and applying equation (35) yields the gradient of the compliance under damage c_d with respect to $\boldsymbol{\eta}^{II}$:

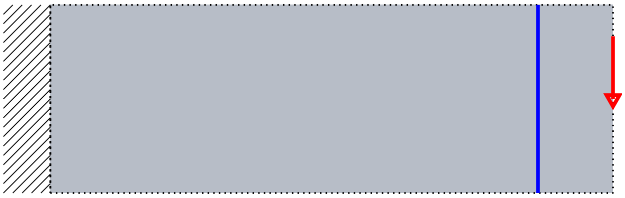


Figure 5: Design domain in 2D with boundary conditions and tip load. Damage is only applied in the region left of the blue line.

$$\frac{dc_d}{d\eta_i^{II}} = \sum_j \frac{\partial c_d}{\partial \bar{\tilde{\varrho}}_j^{II}} \frac{\partial \bar{\tilde{\varrho}}_j^{II}}{\partial \tilde{\varrho}_j^{II}} \frac{\partial \tilde{\varrho}_j^{II}}{\partial \varrho_i^{II}} \frac{\partial \varrho_i^{II}}{\partial \eta_i^{II}} \tag{36}$$

$$= \sum_j \frac{\partial c_d}{\partial \bar{\tilde{\varrho}}_{d,j}^{II}} \frac{\partial \bar{\tilde{\varrho}}_j^{II}}{\partial \tilde{\varrho}_j^{II}} \frac{\partial \tilde{\varrho}_j^{II}}{\partial \varrho_i^{II}} \frac{\partial \varrho_i^{II}}{\partial \eta_i^{II}} \psi_{d,j} \tag{37}$$

5.7. Initial values

The initial design used in stage II is chosen to be close to the preliminary design in the final iteration of stage I. Therefore, the initial values of the shape design variables $\boldsymbol{\eta}^{II}$ are set to a vector with the value η_0^I for each component:

$$\boldsymbol{\eta}^{II} = \eta_0^I \mathbf{I} \tag{38}$$

Hence, for the initial state (iteration 0) of stage II following equality holds (cf. Fig. 4):

$$\boldsymbol{\varrho}^{II}_{initial} = \bar{\boldsymbol{\varrho}}^{I}_{final} \tag{39}$$

However, due to the second filtering step in stage II, slight differences in the projected densities can be observed (slight dilation in sharp corner regions):

$$\bar{\boldsymbol{\varrho}}^{II}_{initial} \neq \bar{\boldsymbol{\varrho}}^{I}_{final} \tag{40}$$

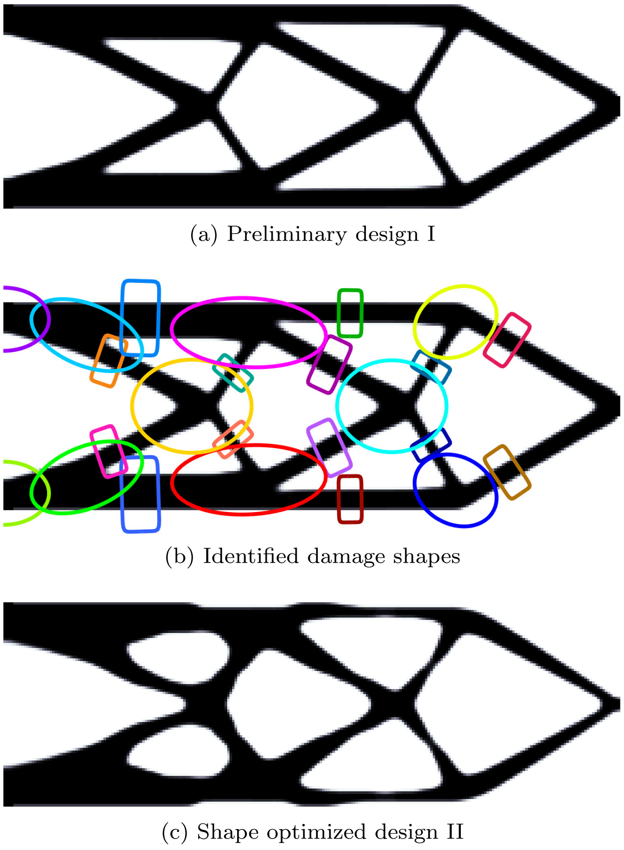
These effects are minimal and very localized, thus the deviations introduced by double filtering of the starting design can be ignored, having in mind that the shape of the part's outline is still strongly dominated by the effect of changing the projection thresholds $\boldsymbol{\eta}^{II}$ throughout the optimization. Additionally, a global volume constraint will make sure the final design does not exceed the maximum allowed volume.

6. Numerical Examples in 2D

For the 2D examples, a design domain of 80 × 240 elements is used. The filter radius is set to $r = 4$. The design domain is pinned on the left side with a transverse unit load acting on the middle node of the right side (see Fig. 5). The loaded end of the beam is excluded from damage considerations (right of the blue line in Fig. 5), meaning that no damage shape may lie in this region.

For all 2D examples, the objective is to minimize the compliance c of the structure for a single load case. The method of moving asymptotes (MMA) by Svanberg (1987) is used as optimization algorithm. For the preliminary design, 200 iterations are performed.

Load path member identification is done using the stress criterion (see Appendix A.1). The damage shapes are obtained by scaling the auxiliary member shapes by a factor of 2. This ensures that the damage shapes are large enough so that any gain of performance in stage II is based on the change of the shape of the individual members and not simply on having circumvented the damage patches. The shapes for the damage on "beam" regions are reduced to slim orthogonal cuts through each beam's midpoint (cf. Fig. 6b).



(a) Preliminary design I

(b) Identified damage shapes

(c) Shape optimized design II

Figure 6: Results for the 2D example of compliance minimization with a global volume constraint.

Table 1: Numerical results for 2D example with global volume constraint.

	Design I	Design II
Volume fraction	40.00%	39.98%
Nominal compliance	211.58	249.40
Worst case compliance	11 263.91	5 486.43

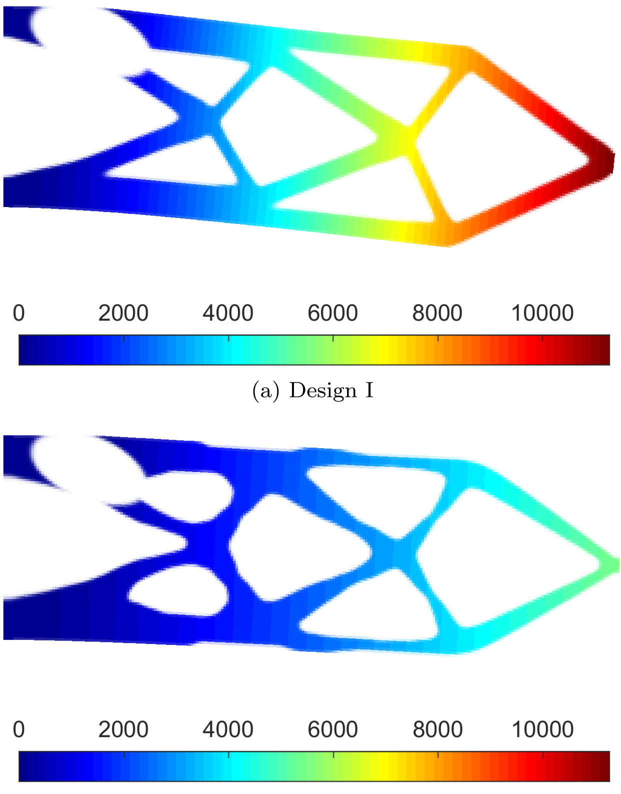
For the shape optimization of stage II, 25 iterations are performed and fast convergence is observed. A global volume constraint is employed in stage II, limiting the volume to be equal or below the volume of the design from stage I.

6.1. Global volume constraint example

A global volume constraint of $k_g = 0.4$ is used in the optimization stage I for the preliminary design. The resulting structure is depicted in Fig. 6a.

Figure 6b shows the outlines of the superellipses that make up the cuts for the individual damage scenarios. In total 14 "beams" and 10 "knots" are automatically identified. The result of the shape optimization in stage II is shown in Fig. 6c. The topology of the structure is conserved.

Numerical values are given in Table 1. The compliance under a worst case damage scenario is reduced to about 49% by the shape optimization. However, a slight penalty of under 18% on the compliance of the undamaged structure is observed. Figure 7 shows the displacements of both designs for their worst case damage.



(a) Design I

(b) Design II

Figure 7: Absolute displacements in the transverse direction of both designs for their worst case damage.

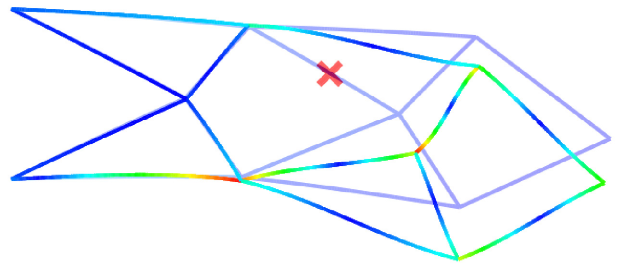
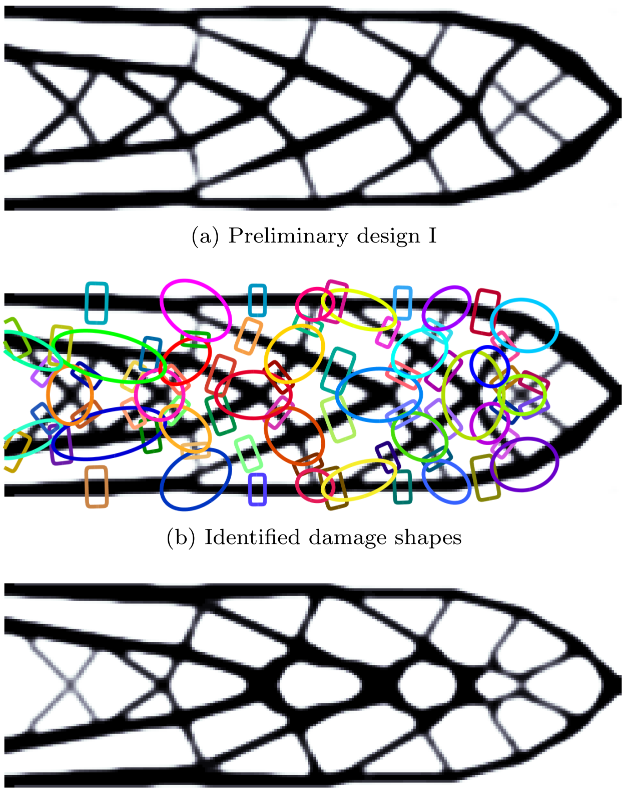


Figure 8: Beam model of the structure with absolute values of cross-sectional moments colored from low values (blue) to high values (red) for the intact (background) and one of the damaged configurations.

The design from stage I consists of beams of constant width since the structure is dominated by tension and compression section forces. In contrast to this, the beams from design II are shaped into beams of varying widths, which means the sections are dominated by bending moments.

To investigate the influence of the bending moments, an additional FE model is set up. The structure is discretized by beam finite elements that each has a thickness assigned to that is extracted from the original density field.

In Fig. 8, the FE model with the absolute values of the cross-sectional moments is shown as a line representation. For the intact structure in the background, the bending moments are low, the load is mostly carried by tension or compression forces.



(a) Preliminary design I

(b) Identified damage shapes

(c) Shape optimized design II

Figure 9: Results for the 2D example of compliance minimization with a local volume constraint.

However, if one of the beams fails, the bending moments become dominant, as shown in the foreground of Fig. 8 for a representative example. The forces are 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 therefore, beams with thin middle sections are obtained.

6.2. Local volume constraint example

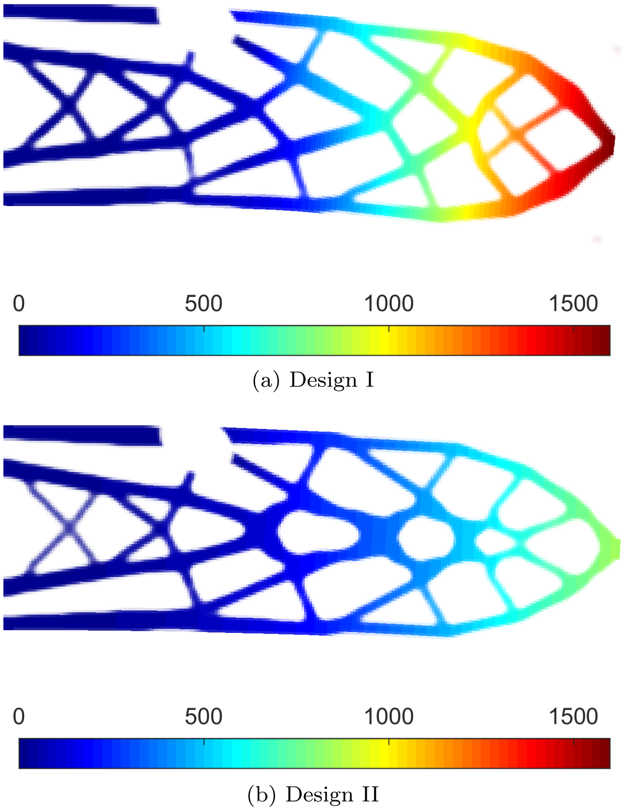
The influence of damages on the structure in Section 6.1 is high since it does not offer redundant load paths. To enforce redundancy, a local volume constraint of $k_l = 0.4$ is used within a control radius of $R = 16$. The redundant load paths from stage I will be conserved in stage II. Therefore, again only the global volume constraint is needed to limit the total volume of the solid phase in stage II.

Figure 9 shows the design from stage I, the identified damage shapes (50 “beams” and 28 “knots”) and the final result from the shape optimization stage II. Numerical values of compliance are given in Table 2.

In this example, the nominal compliance for the undamaged structure has even slightly improved. In stage II only a global volume constraint is active, allowing redistribution of material not allowed for the more restrictive local volume constraint in stage I. Not strictly sticking to the local volume constraint is acceptable here, since applying it in stage I is only a means to obtain more robustness to damage by a redundant topology.

Table 2: Numerical results for 2D example with local volume constraint.

	Design I	Design II
Volume fraction	35.47%	35.46%
Nominal compliance	340.30	330.27
Worst case compliance	1 597.73	847.34



(a) Design I

(b) Design II

Figure 10: Absolute displacements in the transverse direction of both designs for their worst case damage.

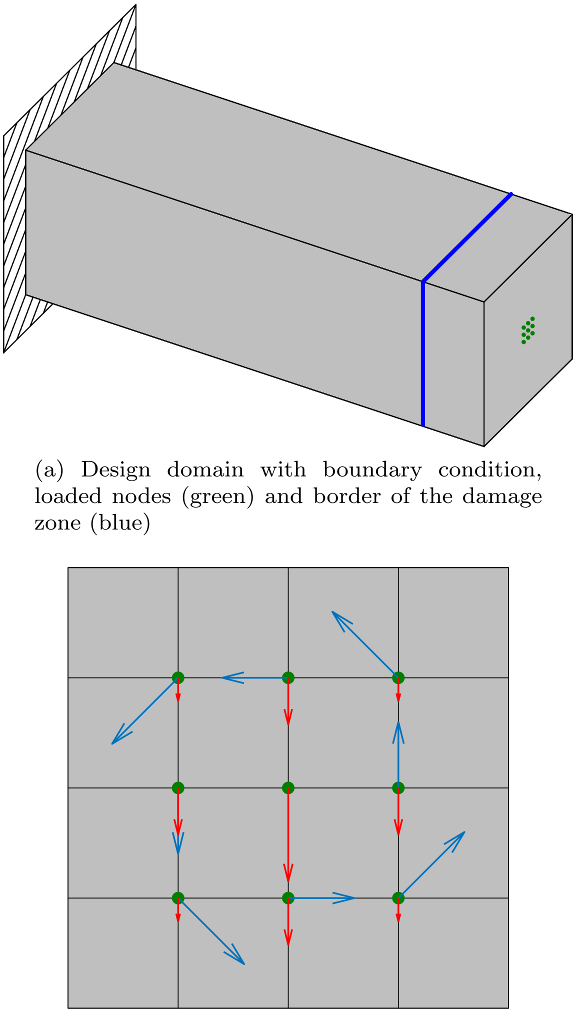
The compliance under a worst case damage is again considerably lowered to about 53%. The according displacements are shown in Fig. 10.

7. Numerical Examples for Lattice-Type 3D Structures

For the 3D examples, a design domain of 20 × 20 × 60 is used. The filter radius is $r = 3$. A lattice-type structure is enforced by using the local volume constraint with low volume fractions in control spheres with a radius of $R = 9$. This procedure breaks shell-like structures down to lattice-type structures which are easier to be automatically identified and partitioned.

The design domain is again pinned on one side. Loads are distributed to a group of 3 × 3 central nodes of the opposite side (see Fig. 11).

For the 3D examples, 300 iterations are performed in stage I and again only 25 iterations in stage II. A global volume constraint in stage II ensures that the shape optimized structure has less or equal volume than the initial structure.



(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

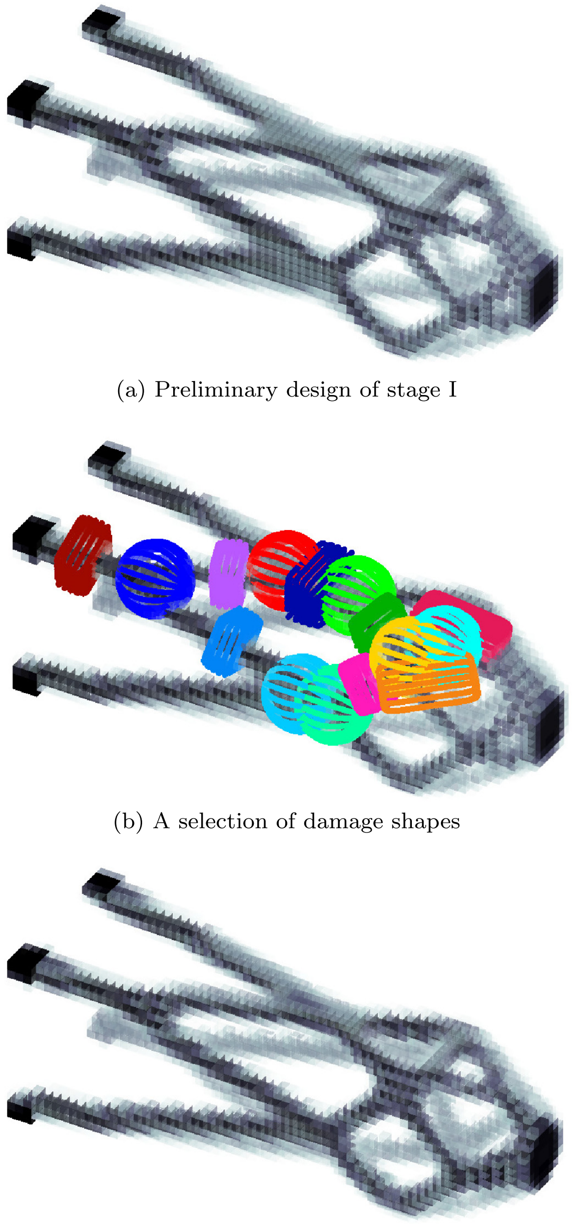
Figure 11: Geometry and loading for the 3D examples. Damage is only applied in the region left of the blue lines in the top figure.

7.1. Lattice-type 3D structure and multiple load cases

The load cases are a torsion load and a downwards pointing transverse tip load with resultant magnitudes of 1 (see Fig. 11b). 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. The local volume constraint is set to $k_l = 0.1$. The resulting structure is shown in Fig. 12a.

Since multiple load cases are considered in the optimization, also the stress states inside the structure will be different between the load cases and the assumption of uniaxially loaded "beam" segments is not met (for details see Appendix A.1). Therefore, the stress criterion cannot be used here. Instead, load paths are identified by volumetric image processing algorithms (see Appendix A.2). For better visibility Fig. 12b only shows the damage shapes for a quarter of the structure. For the full structure 30 "beams" and 21 "knots" are identified.

The result from optimization stage II is shown in Fig. 12c. The change in shape is rather small, but comparing the numerical



(a) Preliminary design of stage I

(b) A selection of damage shapes

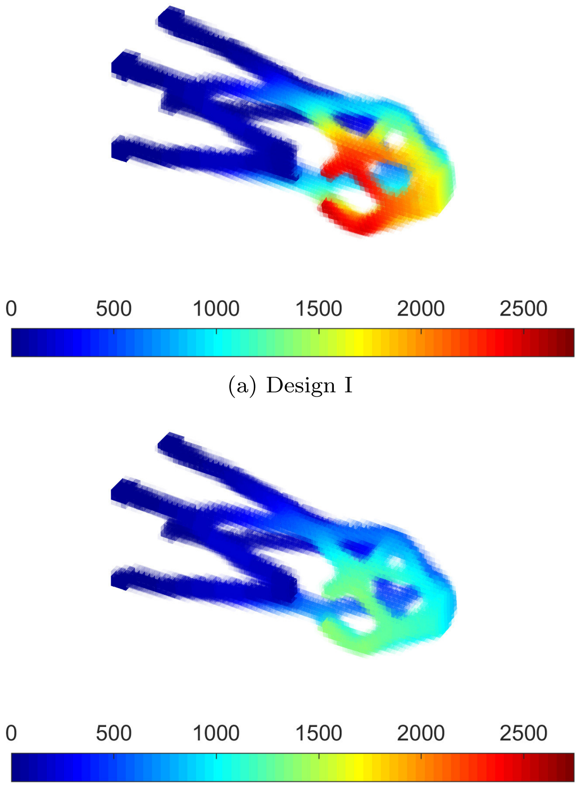
(c) Design of stage II

Figure 12: Results for compliance minimization with a local volume constraint for a structure under torsional loading and transverse load. For better visibility, the beams on the far side are shown with less contrast.

Table 3: Numerical results for 3D example with multiple load cases.

	Design I	Design II
Volume fraction	8.45%	8.44%
Nominal compliance	5.19	5.39
Worst case compliance	23.49	14.64

values in Table 3 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 Fig. 13.



(a) Design I

(b) Design II

Figure 13: Absolute displacements of both designs for their worst case damage under the load case of torsion.

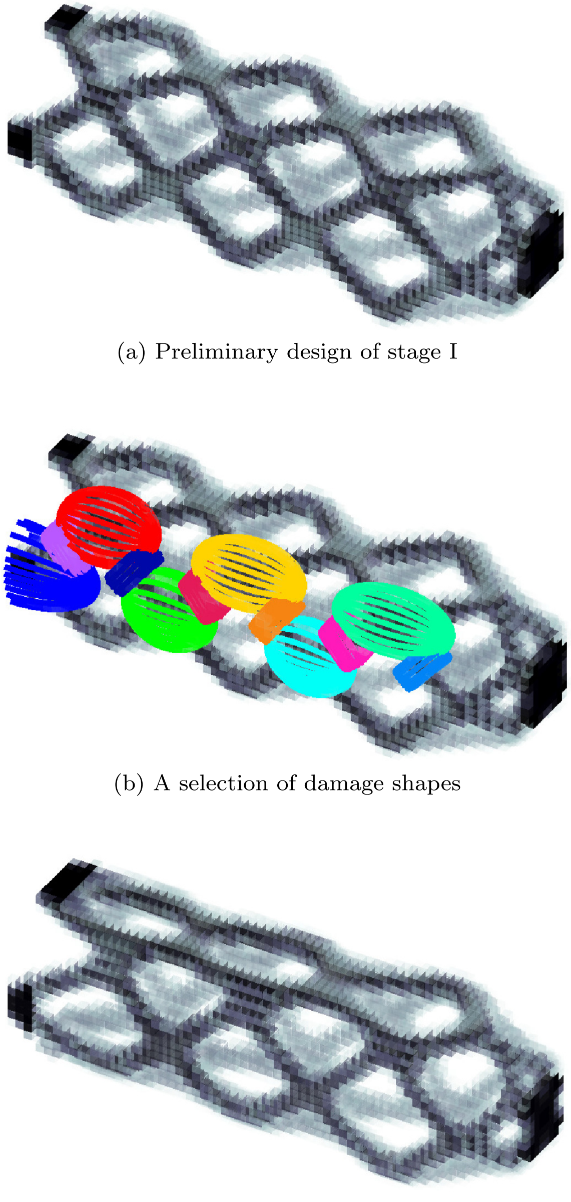
7.2. Lattice-type 3D structure and non-ideal loading

This example demonstrates that more drastic changes in shape are possible in stage II. A structure is 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 Fig. 11b.

A local volume fraction of $k_l = 0.13$ is used, the design from stage I is shown in Fig. 14a. By using the stress criterion, 75 damage shapes (48 "beams" and 27 "knots") are generated. Due to symmetry, only a subset of these shapes is shown in Fig. 14b. In optimization stage II the upper and lower region beams become axially aligned to better absorb the bending introduced by the additional transverse load, as seen in Fig. 14c.

Numerical values for the compliances c are given in Table 4. The performance for the non-ideal loading is increased and the overall 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.

Since the transverse forces are acting in both directions (up and down), two load cases exist for the non-ideal load. Due to symmetry of the design, the compliances for both are the same. The values in Table 4 for the non-ideal loading refer to a single load case, for the mirrored transverse load the same value applies.



(a) Preliminary design of stage I

(b) A selection of damage shapes

(c) Design of stage II

Figure 14: 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 better visibility, the beams on the far side are shown with less contrast.

Table 4: Numerical results for 3D example with misaligned loading.

	Design I	Design II
Volume fraction	11.00%	11.00%
Nominal c, ideal load	0.56	1.00
Worst case c, ideal load	1.49	2.02
Nominal c, non-ideal load[a]	17.66	4.39
Worst case c, non-ideal load[a]	30.66	15.56

[a] Compliance in each direction of non-ideal loading.

8. Discussion of Computational Cost

Table 5 summarizes the overall expected computational cost for a fail-safe optimization of the example problems in 2D and 3D from Sections 6 and 7 using different methods.

The approach of Jansen et al. (2013) is taken as a reference. For this method, a rather large damage patch size of one quarter of the shortest design space dimension is chosen (20 × 20 in the 2D case, 5 × 5 × 5 for 3D). This leads to a total count of 11 041 damage cases for the 2D example and 11 776 for 3D which is the number of FE analyses to be done per single optimization iteration. Assuming that the number of iterations is the same as for the preliminary optimizations in this paper (200 for 2D or 300 for 3D), the total number of FE analyses would be in the millions, rendering the method unfeasible for industrial application.

With the reduction proposed by Zhou and Fleury (2016), using the second coarsest damage density level of 2, still 133 analyses have to be done per iteration for the 2D example and 931 for the 3D example. The total number of analyses would then be in the order of magnitude of 10^4–10^5 for the shown examples. However, these computational savings come at the price that the coarser the damage density level is, the more likely it is that a critical damage location is simply disregarded, as shown by the authors in Ambrozkiewicz and Kriegesmann (2018).

For the proposed method for fail-safe optimization, a constant number of FE analyses for the preliminary design plus a variable number of analyses for each design iteration of the second optimization stage need to be conducted. The total number of analyses is now dependent on the number of identified load paths of the structure and therefore different for every loading scenario considered here. Table 5 reveals that for the shown examples the total number of analyses is in the thousands, independently of the design space being modeled in 2D or 3D.

The density-based shape optimization converges much faster than a topology optimization “from scratch”. Therefore, 25 iterations were sufficient to achieve convergence. 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 here would be lower than with the approach of Zhou and Fleury (2016) and much lower than with the approach of Jansen et al. (2013).

9. Conclusions

A sequential procedure for fail-safe topology and shape optimization has been presented, where the main contribution is the second optimization stage which consists of a density-based shape optimization.

By application to 2D and 3D examples, it has been demonstrated that this density-based shape optimization significantly improves the fail-safe behavior of a structure that originates from a preliminary topology optimization while using the same amount of material.

Compared to topology optimization schemes with explicit fail-safe considerations, the computational effort of the sequential optimization approach is smaller by orders of magnitude which makes it feasible for industrial applications. However, since stage II is a shape optimization, the design freedom is more restricted than for a topology optimization and the result is dependent on the preliminary design of stage I.

Compared to the approach of implicitly increasing damage tolerance only by enforcing redundancy with a local volume constraint, the fail-safe performance of the sequentially optimized

Table 5: Total number of FE analyses.

Example	Method	Damage cases	Total solves
2D	Jansen et al. (2013)	11 041	2 208 200
2D	Zhou, Fleury (2016), level 2	133	26 600
2D (Section 6.1)	Proposed	24	800
2D (Section 6.2)	Proposed	78	2 150
3D	Jansen et al. (2013)	11 776	3 532 800
3D	Zhou, Fleury (2016), level 2	931	279 300
3D (Section 7.1)	Proposed	51	1 575
3D (Section 7.2)	Proposed	75	2 175

structures improves significantly within a few additional iterations of density-based shape optimization. For the 2D example in Section 6.2, the compliance of the worst case damage scenario was further decreased by 47%. Simultaneously, the compliance of the intact structure was also decreased.

This shows that the sequential approach of topology optimization with the local volume constraint and the subsequent density-based shape optimization is an efficient procedure to design fail-safe structures.

Acknowledgements

Financial support of the German Ministry for Economic Affairs and Energy (project REGIS, funding reference 20W1708E) and the German Research Foundation (reference number KR 4914/3-1) is acknowledged.

Conflict of interest statement

Declarations of interest: none.

References

Ambrozkiewicz, O., & Kriegesmann, B., (2018). Adaptive strategies for fail-safe topology optimization. in *EngOpt18-6th International Conference on Engineering Optimization*, Lisbon, Portugal.

Bendsøe, M.P., (1989). Optimal shape design as a material distribution problem. *Structural Optimization*, *1*(4), 193–202.

Bendsøe, M.P., & Sigmund, O., (2004). *Topology optimization theory, methods, and applications* (2nd ed.), Berlin/Heidelberg: Springer-Verlag.

Bourdin, B., (2001). Filters in topology optimization. *International Journal for Numerical Methods in Engineering*, 50(9), 2143–2158.

Bruns, T.E., & Tortorelli, D.A. (2001). Topology optimization of non-linear elastic structures and compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering*, 190(26), 3443–3459.

Hojjat, M., Stavropoulou, E., & Bletzinger, K.-U., (2014). The Vertex Morphing method for node-based shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 268, 494–513.

Jansen, M., Lombaert, G., Schevenels, M., & Sigmund, O., (2013). Topology optimization of fail-safe structures using a simplified local damage model. *Structural and Multidisciplinary Optimization*, 49(4), 657–666.

Kreisselmeier, G., & Steinhauser, R., (1983). Application of vector performance optimization to a robust control loop design for a fighter aircraft. *International Journal of Control*, 37(2), 251–284.

Kriegesmann, B., & Lüdeker, J.K., (2019). Robust compliance topology optimization using the first-order second-moment method. *Structural and Multidisciplinary Optimization*, *60*(1), 269–286.

Lazarov, B.S., Schevenels, M., & Sigmund, O., (2012). Topology optimization with geometric uncertainties by perturbation techniques. *International Journal for Numerical Methods in Engineering*, *90*(11), 1321–1336.

Le, C., Bruns, T., & Tortorelli, D., (2011). A gradient-based, parameter-free approach to shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 200(9), 985–996.

Lüdeker, J.K., & Kriegesmann, B., (2019). Fail-safe optimization of beam structures. *Journal of Computational Design and Engineering*, *6*(3), 260–268.

European Aviation Safety Agency, (2012). Certification specifications for large aeroplanes. Tech. Rep. CS-25, Amendment 12.

Otsu, N., (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 62–66.

Pratt, W.K., (2006). *Digital image processing* (3rd. ed.) , Hoboken, New Jersey: John Wiley & Sons, Inc.

Schevenels, M., Lazarov, B.S., & Sigmund, O., (2011). Robust topology optimization accounting for spatially varying manufacturing errors. *Computer Methods in Applied Mechanics and Engineering*, 200(49–52), 3613–3627.

Sigmund, O., (2009). Manufacturing tolerant topology optimization. *Acta Mechanica Sinica*, 25(2), 227–239.

Sossou, G., Demoly, F., Montavon, G., & Gomes, S., (2018). An additive manufacturing oriented design approach to mechanical assemblies. *Journal of Computational Design and Engineering*, 5(1), 3–18.

Sun, P.F., Arora, J.S., & Haug, E.J., Jr, (1976). Fail-safe optimal design of structures. *Engineering Optimization*, 2(1), 43–53.

Svanberg, K., (1987). The method of moving asymptotes – A new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2), 359–373.

Wang, F., Lazarov, B.S., & Sigmund, O., (2011). On projection methods, convergence and robust formulations in topology optimization. *Structural and Multidisciplinary Optimization*, 43(6), 767–784.

Wu, J., Aage, N., Westermann, R., & Sigmund, O., (2017). Infill optimization for additive manufacturing – Approaching bone-like porous structures. *IEEE Transactions on Visualization and Computer Graphics*, *PP*(99), 1–1.

Zhou, M., & Fleury, R., (2016). Fail-safe topology optimization. *Structural and Multidisciplinary Optimization*, 54(5), 1225–1243.

Appendix A: Automatic Identification of Load-Bearing Members

The presented method for fail-safe optimization uses a set of parametric shapes from which damage scenarios are derived.

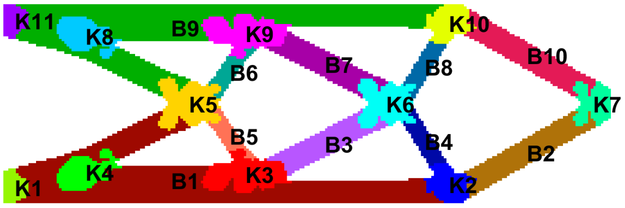


Figure A1: Identified clusters: Beams "B" and knots "K" before splitting.

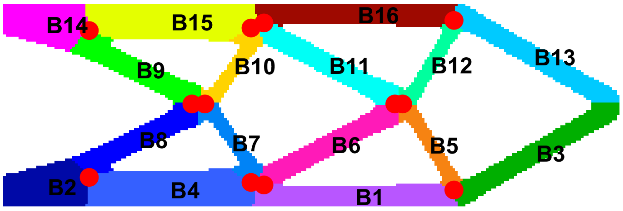


Figure A2: Identified beam clusters by image recognition. Locations of branch points are shown as red dots.

Ideally, these shapes can be extracted by automated algorithms. The authors propose to subdivide the lattice-type result of stage I into "beams" and "knots".

Two methods successfully applied by the authors are briefly presented here: The stress criterion method (Ambrozkiewicz and Kriegesmann 2018) and a method based on image processing algorithms. These methods may be substituted by any other algorithm offering the same functionality or even a manual subdivision can be performed.

A.1. Stress criterion

When using the stress criterion method, no additional simulation is needed and no special algorithms have to be used. The basic 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

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

$$\varphi_n(\boldsymbol{\sigma}) = \frac{4}{\pi}\arctan(\underline{\sigma}/\overline{\sigma}) \tag{A.1}$$

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 2a shows the stress criterion values for the deterministic 2D cantilever beam example.

The classification into beams and knots is then done by choosing a threshold value for the criterion, e.g. based on the empirical cumulative distribution function of the criterion values. For Fig. A1 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 Fig. A1 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 Fig. 2b 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 (Otsu 1979). In this example this was the case for the beams "B1" and "B9" from Fig. A1 which are branched near the left side. In the final result in Fig. 2b, "B12" and "B13" resulted from this splitting procedure.

A.2. Image processing

As an alternative to the stress criterion, methods from image processing can be used for automatic classification. These algorithms work on binary images in 2D or volumetric voxel images in 3D.

For regular FE meshes with a rectangular design space, a binary image can be easily obtained by treating each finite element 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 0/1 binary representation. For irregular meshes, a sampling into an image of finer resolution can be performed.

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 (see Pratt 2006).

The branch points of the skeleton can be extracted via the functions `bwmorph` for 2D or `bwmorph3` for 3D with the argument `'branchpoints'`. Dilated branch points can be directly used as the "knot" regions of the part. However, the actual size of the knot region has to be set manually.

The difference between the skeleton and the branch points gives segmented line representations of the individual "beam" regions. A mapping of each solid element to the nearest skeleton segment or branch point finishes the automatic classification. Applied to the 2D cantilever beam example from Section 6.1 the result from Fig. A2 is obtained, where for knots only the midpoints are displayed.

Appendix B: Calculating Parameters of Auxiliary Shapes from Element Clusters

For each cluster the location of the center point $\mathbf{x}_c$, the matrix $\mathbf{V}$ with unit vectors of the principal axes of the auxiliary shape as well as the corresponding size vector $\boldsymbol{a}$ need to be calculated. This section shows the individual calculation steps if for every cluster a set $\mathbb{S}$ that contains the corresponding element indices is given.

The center point $\mathbf{x}_c$ of the shape is determined by calculating the center of mass of all elements belonging to the cluster:

$$\mathbf{x}_c = \frac{\sum_{i\in\mathbb{S}} \mathbf{x}_i v_i}{\sum_{i\in\mathbb{S}} v_i} \tag{B.1}$$

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 $\mathbf{x}_i$ are concatenated into a matrix $\boldsymbol{X}$ and the covariance matrix $\boldsymbol{\Sigma}$ is calculated:

$$\mathbf{X} = [\mathbf{x}_i, \cdots] \qquad \forall i \in \mathbb{S} \tag{B.2}$$

$$\boldsymbol{\Sigma} = \mathrm{Cov}(\mathbf{X}) \tag{B.3}$$

A Matrix $\mathbf{V}$ with unit vectors of the principal axes as columns is obtained by an eigen decomposition of $\boldsymbol{\Sigma}$:

$$\boldsymbol{\Sigma}\mathbf{V} = \mathbf{V}\boldsymbol{\Lambda} \tag{B.4}$$

$$\mathbf{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3] \tag{B.5}$$

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \tag{B.6}$$

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 be the squared standard deviations:

$$\sigma_i^2 = \lambda_i \tag{B.7}$$

For a 1D uniform distribution with a width of $2a$ the relation to the standard deviation σ is:

$$\sigma^2 = \frac{(2a)^2}{12} \qquad \Leftrightarrow \qquad a = \sqrt{3}\sigma \tag{B.8}$$

This relation can be applied to all principal directions of the shape to obtain the size vector $\boldsymbol{a}$:

$$\boldsymbol{a} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \sqrt{3} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix} \tag{B.9}$$

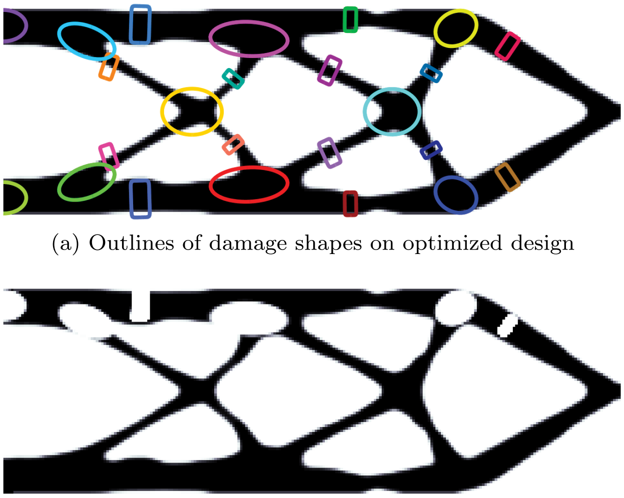
Applying the factor $\sqrt{3}$ from equation (8) as a scaling factor between the standard deviations calculated by equation (7) and the half-axes of the auxiliary shapes stored in $\boldsymbol{a}$ will give a close fit to the original clusters (cf. Fig. 2c).

Appendix C: Scaling of the Damage Shapes

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. Fig. 2c).

If the damage shapes derived from the member shapes (see Section 4.3) are not scaled up, they may easily be circumvented during the fail-safe optimization stage II. Considering the example from Section 6.1 without upscaling the final result from Fig. C1a is obtained which clearly differs from the result shown in Fig. 6c. The most critical damage shapes have been circumvented by material that has been placed just around the damage shape, hinders the corresponding structural member to be cut entirely and therefore jeopardizes the fail-safe optimization.

The numerical value of the compliance for the worst case damage with a damage shape scaling factor of 1 is given in Table C1 and is now much lower than for the example in Section 6.1 with a scale of 2. This is not due to increased robustness to damage but rather due to the incomplete cutting of the individual members. Figure C1b shows some exemplary incomplete cuts in the upper region of the part.



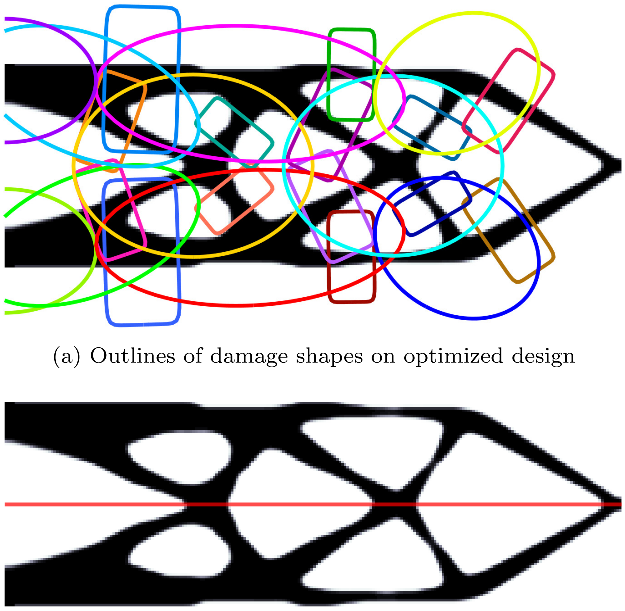
(a) Outlines of damage shapes on optimized design

(b) Some incomplete cuts at potentially critical locations at the top shown simultaneously

Figure C1: Results of the fail-safe optimization for a damage shape scaling of 1.

Table C1: Calculated worst case compliances for optimizations with different damage scaling factors.

Damage scaling factor	Worst case compliance
1 (cf. Fig. C1a)	532.57
2 (cf. Fig. 6b)	5 486.43
4 (cf. Fig. C2a)	5 578.64



(a) Outlines of damage shapes on optimized design

(b) Optimized design for a scaling factor of 4 (top) and 2 (bottom)

Figure C2: Results of the fail-safe optimization for a damage shape scaling of 4.

The other extreme would be a much too high scaling factor. The damage shapes and the result after fail-safe optimization for a scaling factor of 4 are shown in Fig. C2a. Because these overly large damage regions are an even more conservative approach than the ones with the recommended scaling factor of 2 (seen in Fig. 6b), also the worst case compliance listed in Table C1 has increased, but only slightly. The difference in the final structure is minimal when comparing the scaling factors 4 and 2, as shown in Fig. C2b.

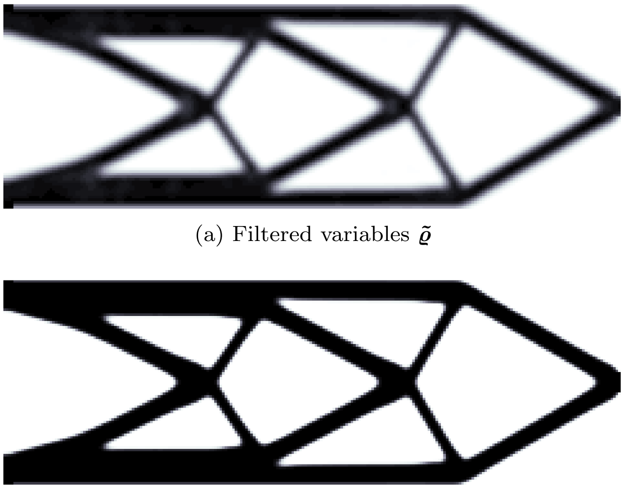
Appendix D: Effect of the Threshold Parameter η

The projection step from equation (7) is applied to the filtered variables $\tilde{\varrho}$. Its purpose is to make interface regions between the solid and void phase more sharp for larger β values.

Figure D1 a shows the filtered variables $\tilde{\varrho}$ and the resulting projected variables $\bar{\varrho}$ when using a uniform $\eta = 0.5$ as the threshold and $\beta = 8$ as the steepness parameter.

Modifying the η value allows for thickness variations of the members of a structure. When altering the threshold value, parts become eroded or dilated. Projecting the filtered variables from Fig. D1 a with uniform values of $\eta = 0.2$ and $\eta = 0.8$ results in the designs shown in Fig. D2.

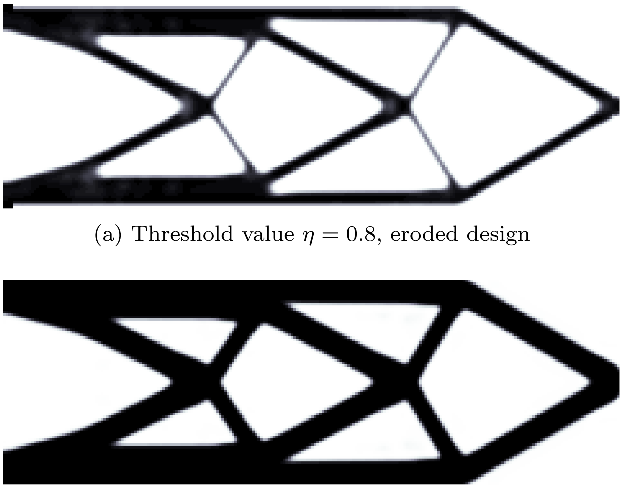
In this paper, a non-uniform projection threshold field $\boldsymbol{\eta}$ is used to apply spatially varying thickness modifications to a part.



(a) Filtered variables $\tilde{\varrho}$

(b) Projected variables $\bar{\varrho}$ with threshold value $\eta = 0.5$

Figure D1: Filtered and projected variables with projection parameter $\beta = 8$.



(a) Threshold value $\eta = 0.8$, eroded design

(b) Threshold value $\eta = 0.2$, dilated design

Figure D2: Effect of uniform variation of η with $\beta = 8$ on the projected variables.