

Article

Multi-Material Topology Optimization on Separate Tetrahedral Meshes with Explicit Design Resolution by Means of Remeshing

Robert Renz * and Albert Albers 

IPEK—Institute of Product Engineering at Karlsruhe Institute of Technology (KIT), Kaiserstr. 10, 76131 Karlsruhe, Germany; sekretariat@ipek.kit.edu

* Correspondence: robert.renz@kit.edu; Tel.: +49-721-608-47165

Abstract: As a method of lightweight design, multi-material design aims to make targeted use of materials in order to reduce CO₂ emissions. In this context, it can be described as one of the product development methods used to meet the challenges of climate change. However, since the design of structures in multi-material design is complex, topology optimization can be used to support the product developer. In this article, a multi-material topology optimization method is developed that combines the Velocity Field Level Set method with the Reconciled Level Set method. Furthermore, the current design is explicitly resolved in each iteration by means of multi-material remeshing. The edge collapse phase in the remeshing process is achieved by applying the producer consumer pattern. The developed method is then validated using known examples from the state of research, and the influence of the parameters of the method on the result is analyzed by means of studies.

Keywords: multi-material; topology optimization; level set; remeshing; mesh adaption



Citation: Renz, R.; Albers, A. Multi-Material Topology Optimization on Separate Tetrahedral Meshes with Explicit Design Resolution by Means of Remeshing. *Algorithms* **2024**, *17*, 460. <https://doi.org/10.3390/a17100460>

Academic Editor: Akemi Galvez Tomida

Received: 11 September 2024

Revised: 7 October 2024

Accepted: 12 October 2024

Published: 16 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Topology optimization is a method that can support the product developer in the lightweight design of structures and is therefore a central tool in product development to reduce CO₂ emissions from products [1]. It is also employed, for example, in the design of heat exchangers [2], to identify the optimal flow path [3] or in the context of medicine for the design of implants [4]. The most widespread method for topology optimization is density-based topology optimization, the development of which began with the publications by Bendsøe and Kikuchi [5] and Bendsøe [6]. In addition to the density-based topology optimization methods, there are many other topology optimization methods. These include, for example, the optimization criteria-based methods [7], the phase field methods [8] or the nature-mimicking evolutionary structural optimization [9,10]. Another topology optimization method is based on the level set method and was primarily developed by Sethian and Wiegmann [11], Wang et al. [12] and Allaire et al. [13]. In this method, the geometry is described during optimization using a discretized level set function. The geometry is evolved by applying the classical shape derivative to determine the front velocity and then solving a Hamilton–Jacobi equation [13].

In addition to the level set-based topology optimization approach just described, there are other variants of using the level set method for topology optimization. An overview is given in [14], and more recently developed methods can be found, for example, in [15,16]. One such method is the Velocity Field Level Set Method (VFLSM) [17]. It allows the use of mathematical programming methods, such as the Method of Moving Asymptotes (MMA) [18] in combination with the level set method, thereby enabling the straightforward integration of several constraints into level set-based topology optimization [17].

Wang and Wang [19] developed the so-called Color Level Set method for the optimization of multi-material systems. For this purpose, they refined a description of multi-material systems known from image processing for level set topology optimization. It allows the description of 2^n materials using n level set functions.

An alternative approach to optimizing multi-material systems is the Reconciled Level Set Method (RCLSM) [20], in which each material is described by its own level set function. The individual level set functions are evolved independently of each other, and any overlaps that arise are subsequently resolved using a correction process [20]. Chen et al. [21] applied the RCLSM in the context of multi-material topology optimization of smart energy harvesters, and Vogiatzis et al. [22] used it for topology optimization of multi-material negative Poisson's ratio metamaterials. Tian et al. [23] combined the RCLSM with the extended level set method to design ferromagnetic soft active structures using topology optimization.

There are three different approaches to represent the current design for the evaluation of the system response during level set-based topology optimization. In the first approach, a density field of a constant background Finite Element (FE) mesh is adjusted based on the values of the level set function. A smoothed Heaviside function is usually used for the adjustment. This variant is similar to density-based topology optimization methods [13]. The second approach applies immersed finite element methods, such as XFEM [24] or CutFEM [25] methods. These use a constant background mesh that is cut by the level set functions. The third approach is to adapt a mesh or remesh the level set function in each iteration [26]. With this approach, the standard Finite Element Method (FEM) can be used.

Allaire et al. [27] developed a level set topology optimization with iterative mesh adaptation. The unstructured mesh is iteratively adapted using classical mesh modification techniques such as edge collapse, vertex smoothing and edge swap. Li et al. [28] combined this mesh adaptation algorithm with the reaction–diffusion equation-based level-set method. It uses the topological derivative and a reaction–diffusion equation instead of the shape derivative and the Hamilton–Jacobi equation [29]. Feppon [2] optimized a heat exchanger also using this iterative mesh adaptation approach.

In this article, a new multi-material topology optimization method for tetrahedral meshes based on RCLSM and VFSLM with iterative remeshing of the multi-material design is developed. The iterative remeshing method is used to resolve the design in each iteration of the optimization. The explicit resolution of the design allows an accurate determination of the sensitivities during the optimization and enables further research that requires the evaluation of quantities and their sensitivities on the interface between the materials with high accuracy. The developed method is validated by two cases. In the developed method, the edge length of the FE mesh can be adjusted. The influence of the edge length and an additional filter radius parameter on the optimized design is investigated in two studies.

Futhermore, in this article a new parallelization strategy based on the producer–consumer pattern for the edge-collapse phase in the iterative remeshing method is developed.

2. Methods

In the following, the boundary value problem of elastostatics, which is used in both single and multi-material cases, is introduced first. For the validation of the developed method, the single-material optimization problem and the VFSLM are then presented. In the next section, the multi-material optimization problem and the RCLSM are introduced. Finally, an overview of the developed optimization method is given and its individual components are explained in detail.

2.1. Boundary Value Problem of Elastostatics

This article considers the quasi-static boundary value problem of elastostatics:

$$\begin{aligned} \nabla \cdot \sigma - f_b &= \mathbf{0} && \text{in } \Omega, \\ \mathbf{u} &= \mathbf{u}_D && \text{on } \zeta_D, \\ \sigma \mathbf{n} &= \mathbf{f}_t && \text{on } \zeta_N. \end{aligned} \quad (1)$$

\mathbf{u} is the displacement field to be identified, and σ is the stress field of the domain Ω . For the Neumann boundary ζ_n with normal vector \mathbf{n} , the force \mathbf{f}_t is specified as a boundary condition. In addition, in this article, the volume forces are assumed to be $\mathbf{f}_b = \mathbf{0}$ and the displacements $\mathbf{u}_D = \mathbf{0}$ on the Dirichlet boundary ζ_D .

To solve the boundary value problem (1), an additional constitutive law is required. For this purpose, Hook’s linear-elastic law is used:

$$\sigma = \tilde{C}[\varepsilon(\mathbf{u})]. \tag{2}$$

The strain tensor $\varepsilon(\mathbf{u})$ is given by $\varepsilon(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)$. In Equation (2), the elasticity tensor \tilde{C} is a function that can be described in the single-material case by $\tilde{C} = \chi_1(\mathbf{x})\mathbb{C}_1$ and, in the multi-material case, by $\tilde{C} = \sum_i \chi_i(\mathbf{x})\mathbb{C}_i$. The indicator functions $\chi_i(\mathbf{x})$ of the i -th material each have the value of 1 if the i -th material is present and 0 otherwise.

Using the constitutive law and the above simplification, the weak form of the boundary value problem can be derived:

$$\int_{\Omega} \varepsilon(\mathbf{u}) \cdot \tilde{C}[\varepsilon(\boldsymbol{\psi})]d\Omega = \int_{\xi_N} \mathbf{f}_t \cdot \boldsymbol{\psi}d\xi \quad \forall \boldsymbol{\psi} \in U_{ad}. \tag{3}$$

Here, U_{ad} denotes the space of the kinematically admissible displacement fields.

If one also introduces the two abbreviations

$$\begin{aligned} a(\mathbf{u}, \boldsymbol{\psi}) &= \int_{\Omega} \varepsilon(\mathbf{u}) \cdot \tilde{C}[\varepsilon(\boldsymbol{\psi})]d\Omega, \\ l(\mathbf{u}, \boldsymbol{\psi}) &= \int_{\xi_N} \mathbf{f}_t \cdot \boldsymbol{\psi}d\xi, \end{aligned} \tag{4}$$

Equation (3) can be written as $a(\mathbf{u}, \boldsymbol{\psi}) = l(\boldsymbol{\psi})$.

2.2. Single-Material

2.2.1. Optimization Problem

The optimization problem for the single-material case is:

$$\begin{aligned} \min_{\boldsymbol{\eta}} w_c \int_{\Omega} \frac{1}{2} \varepsilon(\mathbf{u}) \cdot \mathbb{C}_1[\varepsilon(\mathbf{u})]d\Omega + w_r \int_{\partial\Omega} d\xi \\ \text{s.t. } a(\mathbf{u}, \boldsymbol{\psi}) - l(\boldsymbol{\psi}) = 0, \quad \forall \boldsymbol{\psi} \in U_{ad} \\ f_{V1} \leq \bar{f}_{V1} \\ v^{min} \leq \eta_i \leq v^{max}, \quad i = 1, \dots, n_d \end{aligned} \tag{5}$$

In Equation (5), the weighted objective function of compliance and regularization term is considered, with the two weighting factors w_c and w_r . The latter ensures that a strut-like structure is created and also regularizes the optimization problem [13]. The boundary value problem (3) in weak form exists as an equality constraint, and the displacements $\mathbf{u}(\boldsymbol{\eta})$ must satisfy this constraint at all times. The target volume fraction \bar{f}_{V1} of the material in relation to the volume of the design space V_D is specified as an inequality constraint. The design variables η_i are also constrained by the lower and upper bounds v^{min} and v^{max} . In the case of this article, the bounds are normal velocities, as the design variables in the VFLSM are the normal velocities.

2.2.2. Velocity Field Level Set Method

The description of the part surface during optimization is completed using the level set function $\varphi : \mathbb{R}^3 \rightarrow \mathbb{R}$. An example of a level set function is depicted in Figure 1. The level set function is positive inside the part, zero on the part surface and negative outside the part (in the rest of the design domain). In summary, the following applies:

$$\begin{cases} \varphi(\mathbf{x}) > 0, & \forall \mathbf{x} \in \Omega \quad (\text{part}) \\ \varphi(\mathbf{x}) = 0, & \forall \mathbf{x} \in \partial\Omega \quad (\text{part surface}) \\ \varphi(\mathbf{x}) < 0, & \forall \mathbf{x} \in D \setminus \Omega \end{cases} \tag{6}$$

In this article, a special form of the level set function, the so-called Signed Distance Function (SDF), is used to numerically stabilize the evolution of the surface described below [13]. The function values of the SDF are the signed distances to the surface inside and outside the part. For example, in the part, $\varphi(x) = d(x, \partial\Omega)$ applies, whereby the function d returns the distance between x and $\partial\Omega$. SDFs also have the property that $\|\nabla\varphi(x)\| = 1$ [30].

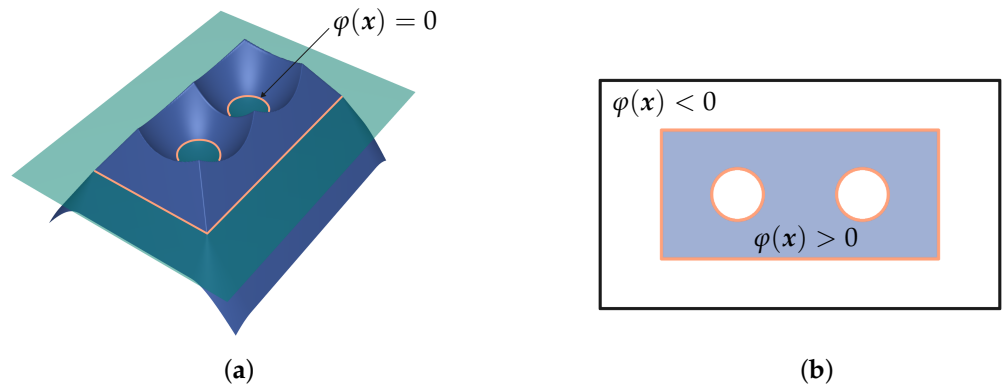


Figure 1. (a) A 2D level set function (blue), which represents a rectangle with two holes. The intersection between the level set function and the plane (green) results in the surface of the rectangle (orange). (b) Representation of the level set function and the corresponding part colored in blue.

The evolution of the level set function and thus the surface ($\varphi(x) = 0$) in a previously given or calculated normal velocity field v^n is described by the Hamilton–Jacobi equation [30]:

$$\frac{\partial\varphi}{\partial t} + v^n \|\nabla\varphi\| = 0. \tag{7}$$

The normal velocity field v^n is linked to the design variables η by linear interpolation in the topology optimization using VFLSM [17]:

$$v^n(x) = \sum_{i=1}^{n_d} b_i(x)\eta_i. \tag{8}$$

Here, $b_i(x)$ is the global basis function of the linear FEM associated with node i and η_i is the respective design variable. When discretizing the normal velocity field, other basis functions can be used in addition to the basis functions from the linear FEM. An overview of this is given in [17].

The design variables are varied by the optimization algorithm based on previously calculated sensitivities. In the VFLSM, the derivation of the sensitivities is based on the shape derivative [17]. The shape derivative of the compliance component of the objective function is based on the simplification described above (no volume forces and no variation in the Neumann and Dirichlet boundary) [13,17]:

$$DJ_c(\Omega)(\theta) = \int_{\partial\Omega} \varepsilon(\mathbf{u}) \cdot \mathbb{C}_1[\varepsilon(\mathbf{u})]\theta \cdot \mathbf{n} d\xi, \tag{9}$$

and the shape derivative of the regularization component [13]:

$$DJ_r(\Omega)(\theta) = \int_{\partial\Omega} \kappa\theta \cdot \mathbf{n} d\xi. \tag{10}$$

The mean curvature κ of the surface $\partial\Omega$ is defined as $\kappa = \nabla \cdot \mathbf{n}$ [13]. Using $v^n = \theta \cdot \mathbf{n}$, the derivative of the objective function with respect to the design variables η_i is given by [17]:

$$\frac{\partial J}{\partial \eta_i} = w_c \int_{\partial\Omega} \varepsilon(\mathbf{u}) \cdot \mathbb{C}_1[\varepsilon(\mathbf{u})]b_i d\xi + w_r \int_{\partial\Omega} \kappa b_i d\xi, \quad i = 1, \dots, n_d. \tag{11}$$

Similarly, the following derivative for the volume constraint can be derived [17]:

$$\frac{\partial f_{V1}}{\partial \eta_i} = -\frac{1}{V_D} \int_{\partial\Omega} b_i d\xi, \quad i = 1, \dots, n_d. \tag{12}$$

2.3. Multi-Material

2.3.1. Optimization Problem

The optimization problem for the multi-material case with two materials is as follows:

$$\begin{aligned} & \min_{\eta} w_c \int_{\Omega} \frac{1}{2} \boldsymbol{\varepsilon}(\mathbf{u}) \cdot \check{\mathbb{C}}[\boldsymbol{\varepsilon}(\mathbf{u})] d\Omega + w_{ri} \int_{\Gamma} d\Gamma + w_{rs} \int_{\partial\Omega} d\xi \\ \text{s.t. } & a(\mathbf{u}, \boldsymbol{\psi}) - l(\boldsymbol{\psi}) = 0, \quad \forall \boldsymbol{\psi} \in U_{ad} \\ & f_{V1} \leq \bar{f}_{V1} \\ & f_{V2} \leq \bar{f}_{V2} \\ & v^{min} \leq \eta_i \leq v^{max}, \quad i = 1, \dots, n_d \end{aligned} \tag{13}$$

For the regularization in the multi-material case, the area of the interface Γ between the two materials and their surface $\partial\Omega$ is used. The former regularization ensures that the optimization problem is solvable, and the latter ensures strut-like structures in the result. The weighting is performed analogously to the single-material case with the weighting factors w_c , w_{ri} and w_{rs} . In addition, there is another volume constraint for the volume fraction of the second material f_{V2} in the multi-material case compared to the single-material case.

2.3.2. Reconciled Level Set Method

This article applies the RCLSM [20] to describe the multi-material case. Each material is represented with its own level set function in the RCLSM. Here, these are the two level set functions φ_1 and φ_2 (see Figure 2).

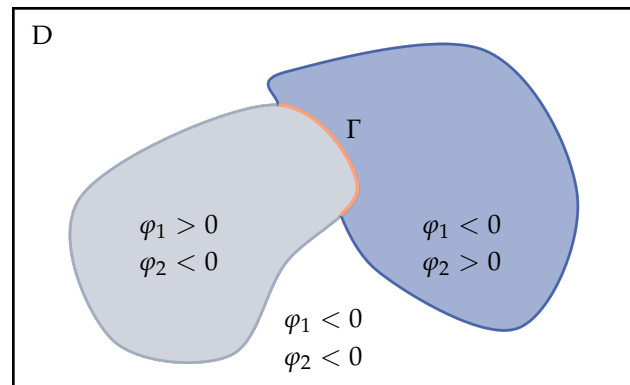


Figure 2. Reconciled level set method: The level set functions $\varphi_1(x)$ (gray) and $\varphi_2(x)$ (blue) each represent the boundaries of a material ($\varphi_i(x) = 0$) in the design domain D . Γ (orange) is the interface between the two materials.

The two level set functions are developed separately using the Hamilton–Jacobi Equation (7). The sensitivities from the single-material case are used to determine the normal velocity fields required for this (see Equations (11) and (12)). This is feasible because the RCLSM is coupled with the VFLSM in this article. However, the elasticity tensor associated with the material (or the level set function) must be used for the compliance component of the sensitivity (e.g., for $\varphi_2(x)$ \mathbb{C}_2).

In the multi-material case, the design variable vector takes the form $\boldsymbol{\eta} = [\eta_1^1, \dots, \eta_{n_d}^1, \eta_1^2, \dots, \eta_{n_d}^2]^T$.

Since the two level set functions are developed independently in RCLSM, the individual level set functions may overlap during evolution (see Figure 3). However, this problem can be solved using the Merriman–Bence–Osher (MBO) operator [20]:

$$\varphi_i = \frac{1}{2}(\hat{\varphi}_i - \max_{i \neq j} \hat{\varphi}_j). \quad (14)$$

This means that the RCLSM works like a predictor–corrector method. In the first step, the level set functions are further evolved in the respective existing velocity field without considering the overlap (predictor step). This results in the temporary level set functions $\hat{\varphi}_1$ and $\hat{\varphi}_2$ (predictor step). The overlaps are then removed using Equation (14) (correction step). The described method is also depicted in Figure 3.

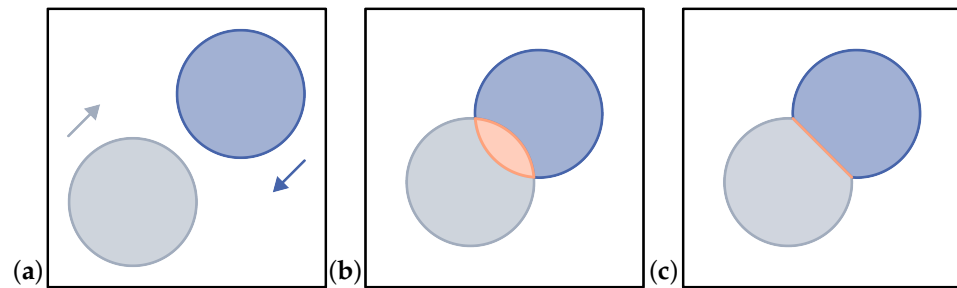


Figure 3. Visualization of the MBO operator using two level set functions $\varphi_1(x)$ (gray) and $\varphi_2(x)$ (blue) moving towards each other based on [22]. (a) $\varphi_1(x)$ and $\varphi_2(x)$ are moving towards each other; there is no overlap. (b) $\varphi_1(x)$ and $\varphi_2(x)$ overlap in the orange area. (c) Correction of $\varphi_1(x)$ and $\varphi_2(x)$ by applying the MBO operator; there is no longer any overlap.

2.4. Optimization Method

The optimization loop of the developed method is shown in Figure 4. At the beginning of each iteration, the current design, represented by the level set functions on the level set mesh, is remeshed with tetrahedra using the method described below in Section 2.4.5. The result of the remeshing is the FE mesh of the current design. This is then used to determine the system response using linear FEM and to calculate the sensitivities of the objective function and the volume constraints (see Section 2.2.2). The calculated sensitivities serve as input for the MMA [18], which updates the design variables (normal velocities). The implementation of the MMA follows [31]. Since the normal velocities are only non-zero in the immediate vicinity (on the surface) of the level set functions, they are extrapolated to the entire design domain (constant level set mesh) using the velocity extension method described in Section 2.4.1. To stabilize the optimization method, the extrapolated normal velocities are then filtered using the Helmholtz equation (see Section 2.4.2). The filtered normal velocities are then used to evolve the level set functions (see Section 2.4.3). The evolved level set functions are then corrected using the MBO operator, and subsequently, their SDF property is restored using redistancing (see Section 2.4.4).



Figure 4. Structure of the optimization loop with the individual components.

2.4.1. Velocity Extension

The normal velocity field on the entire level set mesh is required for each level set function to be evolved. However, since the sensitivities are only non-zero in the vicinity of the zero level sets (surfaces) and consequently the normal velocities (design variables) updated by the optimizer are also non-zero, these must be extrapolated to the rest of the design domain (LS mesh).

For this purpose, a method is developed in this article with an analogous approach to the redistancing method described in [32]. Extrapolation is carried out in the developed method in a two-stage process, which is outlined in Figure 5. In the initialization phase, the surface described by the zero level set ($\varphi(x) = 0$) is sampled. For this, the surface elements (triangles) belonging to the zero level set are derived from the FE mesh using the material IDs. The values of the normal velocities are then determined for each node of the surface mesh using linear interpolation with the FE basis functions from the LS mesh.

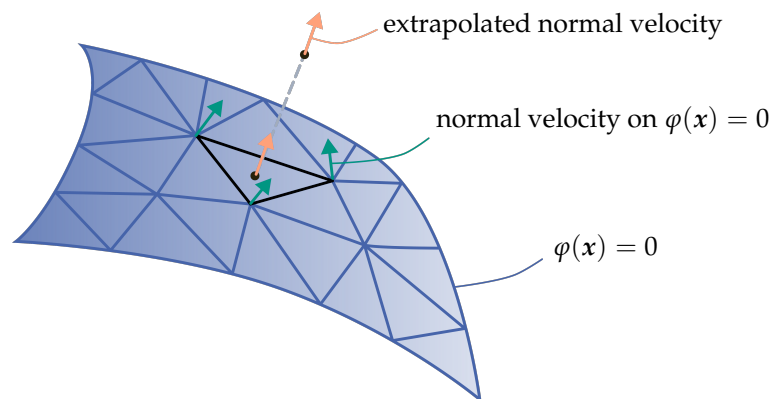


Figure 5. Schematic representation of the velocity extension from the zero level set (blue) to the target vertex based on [33]. For this purpose, the pole of the target vertex is determined on the zero level set, and then, the normal velocity of the target vertex (orange) is calculated from the normal velocities of the triangle (green) using linear interpolation.

The result of the initialization phase is a point cloud of the surface $C_s = \{x_1, x_2, \dots, x_{n_s}\} \subset \mathbb{R}^3$ analogous to [32] with interpolated values and their adjacent triangles. The point cloud is also used to create a search data structure (kd tree).

In the extrapolation phase, the projection of each vertex of the level set mesh (design domain) is identified on the discretized level set surface mesh. The search data structure

and the list of adjacency triangles are used for this purpose. At the identified point, the normal velocity is interpolated using linear interpolation and assigned to the original node in the level set mesh.

The two steps described can be parallelized straight-forwardly by means of multi-threading since both the interpolation step in the initialization phase for each vertex of the surface mesh and the extrapolation step for each vertex of the LS mesh can be carried out independently of each other.

2.4.2. Velocity Filtering

After the normal velocities have been extrapolated into the entire design domain, a filter step is then performed. The filter is applied separately for the two normal velocity fields. The aim of filtering is to smooth the velocity field in order to stabilize the optimization process. The filter used is the one described in [34], which is based on the Helmholtz-PDE and is well known in the field of homogenization-based topology optimization. The filter equation (Helmholtz PDE) is [34]:

$$-r_f^2 \nabla^2 \check{v}^n + \check{v}^n = v^n. \tag{15}$$

\check{v}^n is the filtered normal velocity field that results as the solution of Equation (15). The filter parameter r_f is a length parameter that describes the size of the filter mask but not its explicit radius [34].

To solve Equation (15), its weak form is generated. The building blocks from the evaluation of the system response are used for the evaluation of the weak form. The weak form is solved on the LS mesh with homogeneous Neumann boundary conditions $\frac{\partial \check{v}^n}{\partial n} = 0$ and the unfiltered normal velocity field v^n as the right-hand side [34].

The advantages of this filter approach over explicit filters are the reuse of the building blocks, including their parallelization and the automatic consideration of non-convex design spaces (holes, undercuts) [34].

2.4.3. Advection

In this section, the solution of the Hamilton–Jacobi Equation (7) on tetrahedral meshes using the finite difference method is presented. The method described in [35] is used for this purpose.

The Hamilton–Jacobi Equation (7) can be written as [35]:

$$\frac{\partial \varphi}{\partial t} + H(\nabla \varphi, \mathbf{x}) = 0 \quad \text{with} \quad H(\nabla \varphi, \mathbf{x}) = v^n \|\nabla \varphi(\mathbf{x})\|. \tag{16}$$

The numerical solution of the equation using finite differences requires a stable and accurate upwind approximation of the Hamiltonian $H(\nabla \varphi, \mathbf{x})$ [35]. In the method for tetrahedral meshes developed in [35], the approximation using an implicit mesh dual to the level set mesh (cf. Figure 6) is applied as follows:

$$H_{+\alpha}^u = v_\alpha \frac{\sum_{c \in \alpha} \gamma_c \|\nabla \varphi_{z(c)}\|}{\sum_{c \in \alpha} \gamma_c}. \tag{17}$$

Here, the upwind approximation $H_{+\alpha}^u$ is formed by weighting the cell-centered values H_z surrounding the node α with the corner weights γ_c . These are calculated as [35]:

$$\gamma_c = \max(0, \frac{\mathbf{v}_c}{\|\mathbf{v}_c\|} \cdot \mathbf{a}_c). \tag{18}$$

In this case, \mathbf{a}_c are the cell corner unit normals, which are derived by summation and normalization from the outward corner surface area normals $\mathbf{A}_c = \sum_{i \in c} \mathbf{A}_i = -\sum_{i \in c} \mathbf{S}_i$. The corner velocities in the case of normal advection are given as $\mathbf{v}_c = v_\alpha \frac{\nabla \varphi_{z(c)}}{\|\nabla \varphi_{z(c)}\|}$. For

the calculation of the corner velocities and the upwind approximation, the cell centered gradients are still required:

$$\nabla \varphi_z = \frac{1}{V_z} \sum_{i \in z} A_i \varphi_{p(i)}. \tag{19}$$

Here, V_z denotes the volume of the current tetrahedron of the level set mesh, $i \in z$ means that all surfaces of the tetrahedron are summed up and $\varphi_{p(i)}$ are the level set values associated with the surfaces A_i [35].

The integration in time is carried out using the two-stage Runge–Kutta method. The integration is performed from the current time n to time $n + 1$, using a time step of Δt . It is referred to as two-stage because the intermediate step $n + \frac{1}{2}$ is calculated for the calculation of the $n + 1$ time step [35].

$$\begin{aligned} \varphi_\alpha^{n+\frac{1}{2}} &= \varphi_\alpha^n - \frac{\Delta t}{2} H_{+\alpha}^n, \\ \varphi_\alpha^{n+1} &= \varphi_\alpha^n - \Delta t H_{+\alpha}^{n+\frac{1}{2}}. \end{aligned} \tag{20}$$

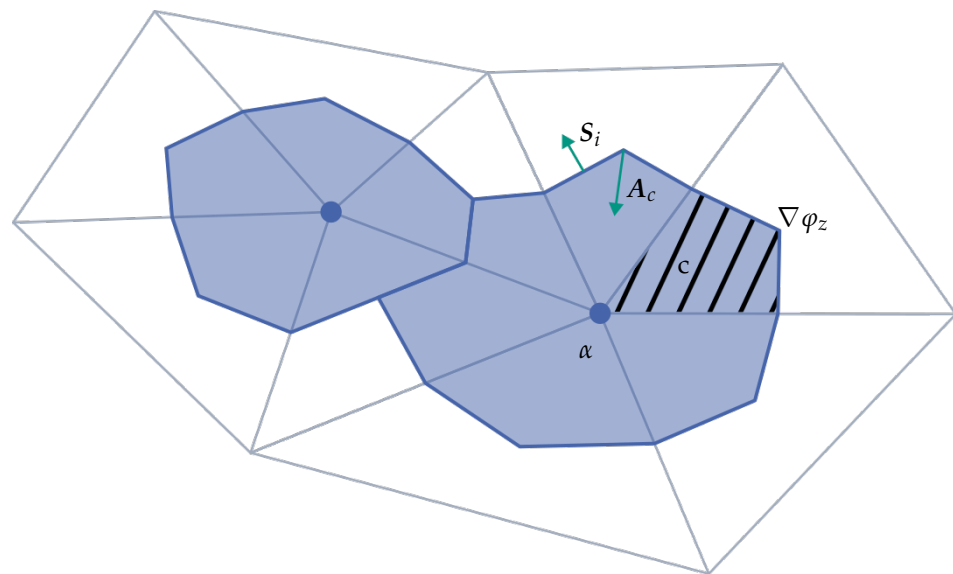


Figure 6. Level set mesh (gray) and the dual mesh with corresponding cells (blue) based on [35]. The dual mesh is used to develop an accurate upwind approximation of the Hamiltonian. A cellcorner c is shaded black.

2.4.4. Redistancing

In the progress of the optimization, the signed distance property ($\|\nabla \varphi(x)\| = 1$) of the level set functions disappears due to the evolution by means of the Hamilton–Jacobi Equation (7) and after application of the MBO operator (14). However, this is essential for the stability of the optimization process, which is why it is restored by means of redistancing [13,21].

This article uses a two-step explicit method for redistancing. In the first step, the level set function is discretized using the Marching Tetrahedra (MT) method. For this purpose, all tetrahedra of the level set mesh are looped. For each tetrahedron, the intersection points of the edges with the zero level set function are determined and the respective MT case is identified based on the edge intersection pattern (see Figure 7) [36].

In the second step, the closest point on the discretized surfaces and the distance to this point are determined for each vertex of the level set mesh. The distance calculation is accelerated using an AABB tree. In addition to the distance, the sign is required, which is determined using the fast winding number [37].

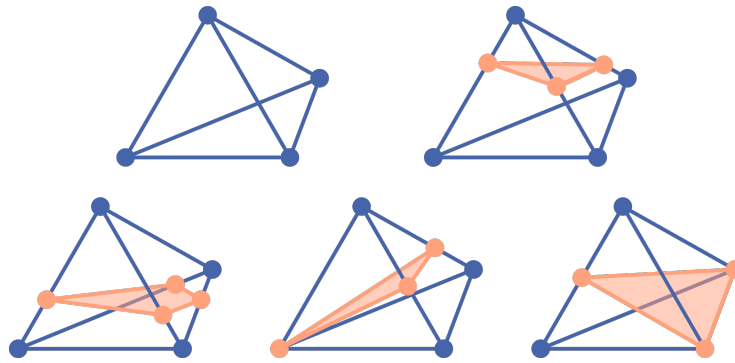


Figure 7. Five unique marching tetrahedra cases based on [36]. The resulting discretized interface consisting of triangles is shown in orange.

2.4.5. Remeshing

This section describes the remeshing method and its implementation for resolving the current design in each iteration. A copy of the LS mesh serves as the starting point. Analogous to the redistancing method, all tetrahedra are looped and split based on the MT cases (see Figure 7) and the individual tetrahedra are assigned the respective material ID. The resulting mesh is then improved using the method described in [38]. The method consists of the following four iterative phases [38]:

- Split all edges with $l_e > l_{e_{max}}$,
- Collapse edges $l_e < l_{e_{min}}$,
- Swap edges of elements with $q < q_{min}$,
- Smooth all vertices.

In the first phase, edges with a length greater than $l_{e_{max}} = \frac{4l_{tar}}{3}$ are split by creating an additional vertex in the middle of the edge to be split. All tetrahedra adjacent to the updated edge are updated. In the second phase, all edges with a length less than $l_{e_{min}} = \frac{4l_{tar}}{5}$ are collected, and then, all permissible collapses are performed. The first two phases of the remeshing method primarily ensure that the edge length converges towards the target edge length l_{tar} . In the third phase, all tetrahedra whose quality q is less than q_{min} are identified [38]. The volume-length measure [39] is applied as a quality measure. For the adjacent edges, all possible edge swaps are tested and the one with the best possible quality is performed. The edge swap phase is primarily used to improve the tetrahedral quality. In the fourth phase, all vertices are repositioned in a hierarchical smoothing. For this purpose, the vertices are divided into volume and boundary vertices and successively smoothed. The classification of the vertices is explained in more detail below [38]. Smoothing the boundary vertices involves a projection step onto the surface, where we project onto the level set functions using Newton's method instead of onto a moving least square-based geometric definition as in [38].

In the phases listed above, many operations occur that are not permitted because they are not feature-preserving or because inverted tetrahedra are created during execution. For the former, a concise set of rules was established in [38], which is based on a classification of vertices, edges and tetrahedra as well as associated conditions. The classification uses the number of labels of the adjacent tetrahedra (here: #material ids). For vertices, for example, if there is exactly one material in the adjacent tetrahedra, it is a volume vertex and if there is more than one material, it is a boundary vertex (see Figure 8a). The classification then allows rules to be defined that describe which mesh modifications are permitted. This is illustrated by the example of an edge collapse in Figure 8b,c [38].

The four phases described above are followed by a quality improvement phase, in which the last remaining tetrahedra with poor quality are eliminated [38]. At this point, we also use the method described in [40], which classifies the bad tetrahedra and then applies targeted operations depending on the type of tetrahedron.

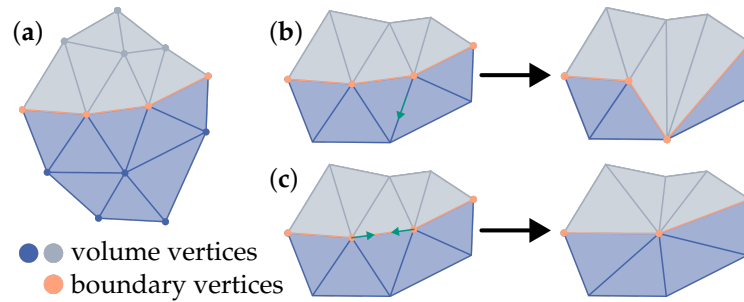


Figure 8. (a) Mesh with classification of vertices into volume and boundary vertices based on [38]. (b) Invalid edge collapse since the collapse does not preserve the interface position (non-feature preserving). (c) Valid edge collapse since the collapse preserves the interface (feature preserving).

For the implementation of the mesh modification (edge collapses, edge swap, etc.), we use the unique cavity operator from [41]. This allows the mesh modifications required for the phases described above to be represented using the same operator and, therefore, the same code. This can significantly reduce the implementation and maintenance effort for the mesh modification [41].

The different operations can be implemented by different initializations and subsequent filling of the cavity [41]. Figure 9 shows an edge collapse and an edge swap using a 2D triangle mesh. For the edge collapse $A \rightarrow B$ (Figure 9a), all elements (tetrahedrons or triangles) are added to the cavity adjacent to vertex A for the initialization, after which they are all deleted. The adapted mesh is obtained by connecting all surfaces of the cavity that do not contain vertex B with target vertex B .

In order to obtain a robust remeshing method, the first step is to test whether inverted tetrahedra are created during the filling process of the cavity operator. The filter-based orientation predicate described in [42] is used for this purpose. Secondly, when initially cutting the mesh with the level set functions based on the MT cases, a vertex can be zero, negative or positive instead of only positive or negative.

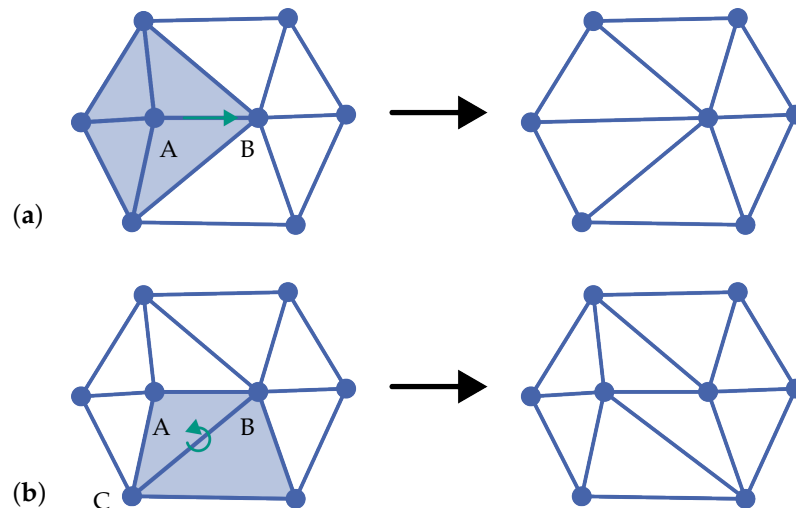


Figure 9. Examples of cavity operations. (a) Edge collapse: the cavity is initialized with all adjacent elements (triangles or tetrahedra) of vertex A , and vertex B serves as the target vertex for the meshing. (b) Edge swap: the cavity is initialized with all elements adjacent to edge BC , and node A serves as the target node for meshing.

A disadvantage of the cavity operator is the somewhat slower performance compared to explicitly implemented mesh operators. However, this disadvantage can be remedied by thread-based parallelization. In this paper, the vertex smoothing and the edge-collapse phase are parallelized. For the vertex smoothing phase, a heuristic graph coloring is used to

identify independent cavities. The vertices of each color are then processed using `#threads`. After the vertices of a color have been smoothed, the threads are synchronized [43].

The edge collapse phase is parallelized using the producer consumer pattern. We use `#threads-1` producer and 1 consumer. After the edges with too short a length have been identified, the producers test all the preconditions described above and then attempt to perform the collapse virtually. If a collapse is valid, it is entered in a queue. The consumer removes the pre-selected collapses and executes them with renewed preliminary checks. The latter is necessary, as the mesh in the vicinity of the edge may have changed in the time between insertion into the queue and execution of the collapse due to other collapses executed by the worker, meaning that the collapse may no longer be valid. The parallelization strategy is based on the fact that only about 10% of the edge collapses per iteration are valid and can be executed.

The cavity operator described only works inside the volume and not at the boundary. Since no extra operator is implemented for the boundary region and the method described requires a convex region, a mesh preparation step is necessary [38]. In the first step, the convex hull of the region is determined and an offset is then formed from this using the method described in [44]. The resulting volume is filled with tetrahedra. We use the `fTetWild` algorithm [45] for this. This preparation now makes it possible to apply the cavity operator described above to the entire area, as the filled offset means that the surface tetrahedra are surrounded by elements (see Figure 10) [38].

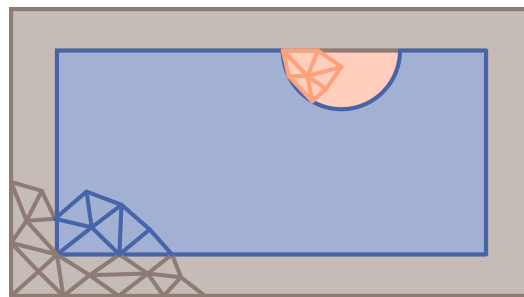


Figure 10. Pre-processing step for remeshing the blue design area. Determination of the convex hull (orange), offset calculation of the convex hull (gray) with subsequent meshing.

3. Numerical Experiments

The following section first presents the two cases that are used in the article to illustrate and validate the developed optimization method. In addition, a summary of the parameters that apply to all numerical examples is then provided. The parameters apply for as long as not directly stated otherwise. This is followed by two single-material optimization examples to validate the single-material case and then two multi-material examples to validate the multi-material case.

3.1. Cases

The first case is a cube with side length 80 with chamfered edges on the top (see Figure 11). The case serves as an example for almost arbitrary design spaces of the optimization method. However, it can be compared at the same time with the classic example of the box without chamfering from the literature. The box is supported at four bearing points on the underside. The displacement of the vertices of one of these supports is restricted in all three spatial directions ($u_x = u_y = u_z = 0$). The vertices of the other supports can move freely in the plane but not in the z -direction ($u_z = 0$). The square load application and constraint surfaces are slightly protruding. The first element layer adjacent to the surfaces is always considered as a non-design space in the optimization and cannot be modified.

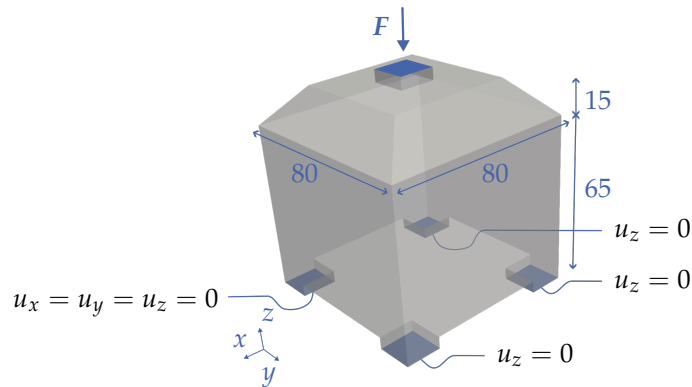


Figure 11. Design space of a box with chamfered edges, which is supported at four locations in the z -plane and has a load application location on the surface with a force in the negative z -direction in the amount of F .

The second case is a cantilever clamped on one side (see Figure 12), which is subjected to a force F in the negative z -direction on the front surface. Two square surfaces with a side length of 15 length units are used for clamping. The vertices of the upper surface are completely fixed in all three spatial directions ($u_x = u_y = u_z = 0$) and the vertices of the lower surface are only fixed in the x and y directions ($u_x = u_y = 0$). Only half of the design domain is considered with an additional symmetry boundary condition so that the result of the optimization is a symmetrical design. The dimensions of the design domain can be taken from Figure 12. As with the box, the load application and constraint surfaces are slightly protruding, and the first element layer is a non-design space.

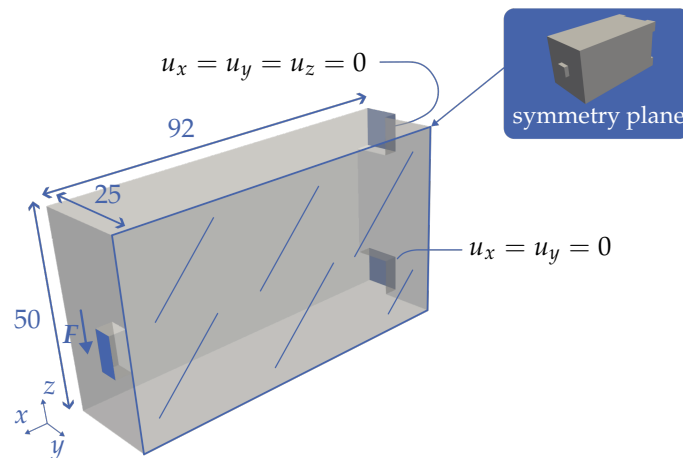


Figure 12. Symmetrical design space of the cantilever beam with a fixed support ($u_x = u_y = u_z = 0$), a floating support ($u_x = u_y = 0$) and a load application point on the front side with a force in the negative z -direction in the amount of F .

For both the single and multi-material cases, $v_{min} = -0.8$ and $v_{max} = 0.8$ are used. A normal velocity of -0.2 is specified for the non-design spaces described above. The filter parameter is selected as 1.25 times the average edge length, and the target edge length is $l_{tar} = 2.0$. The target volume fraction is set to 15% of the design space. In the multi-material case, this is divided equally between the two materials so that $f_{V1} \leq 0.075$ and $f_{V2} \leq 0.075$ applies.

To determine the convergence of the optimization, a convergence criterion is evaluated after the optimization has been performed. This takes into account the five previous iterations in the sense of a moving average and uses this filtered value to calculate the per-

centage change between two points. In the single-material case, the limit for the percentage change is set at 1×10^{-4} and 1×10^{-3} in the multi-material case.

3.2. Single-Material

For the two single-material cases, we used the parameters described below in addition to those listed above. A Young's modulus of $E = 210,000$ and $\nu = 0.3$ is used for the material. The weighting is selected as $w_c = 0.6$, $w_r = 0.4$ for the box case and $w_c = 0.5$, $w_r = 0.5$ for the cantilever. The regularization term is activated after iteration 30 to ensure that it does not dominate the optimization at the beginning. The level set function is redistanced every 5 iterations. The c parameter of the MMA algorithm is increased from a starting value of 1.0 to 100.0 within 150 iterations by means of a linear ramp. The latter results in a slow and smooth volume reduction, which gives the optimization method sufficient time to shape the design according to the objective function.

Figure 13a shows the initial hole structure used to initiate the level set function, and Figure 13b shows the corresponding optimization result at convergence at iteration 107 of the box in the single-material case. It is evident that four main struts are formed to transfer the loads during the optimization process. These transfer the load directly from the point of application to the support points. In addition, cross struts develop in the form of a cross, which prevent the three floating supports from sliding sideways. The optimization result is in line with the state of the research (see [46]). However, it should be noted that depending on the optimization parameter (e.g., the c parameter of the MMA algorithm), the optimization sometimes reaches a different local optimum. This results in a rectangular cross strut structure between the four main struts instead of the cross-shaped cross strut structure.



Figure 13. (a) Initial hole structure of the box in the single-material case. (b) Optimization result for the box at iteration 107.

In addition to comparing the optimization results with the current state of research, further validation activities were carried out. Unit tests were developed for the individual program building blocks. In addition, tests were implemented and then visualized and compared using Paraview. Furthermore, the FEM was compared with the validated FEM software Abaqus (<https://www.3ds.com/products/simulia/abaqus>). Abaqus was also used to validate the compliance objective function and the volume constraint.

Figure 14a,b show the filtered history of the objective function and the volume constraint, respectively. In order to filter out the numerical fluctuation introduced by the remeshing process, a moving-average filter is applied that takes into account the five past iterations. The history of the volume constraint shows the uniform decrease in volume over time.

Figure 15a shows the initial hole structure of the cantilever in the single-material case and Figure 15b the resulting optimization result at iteration 60. The corresponding curves of the filtered objective function and volume constraint are shown in Figure 16a and Figure 16b, respectively.

The optimization result shows the usual truss structure consisting of triangles and agrees with the results from the state-of-the-art research (see [13]).

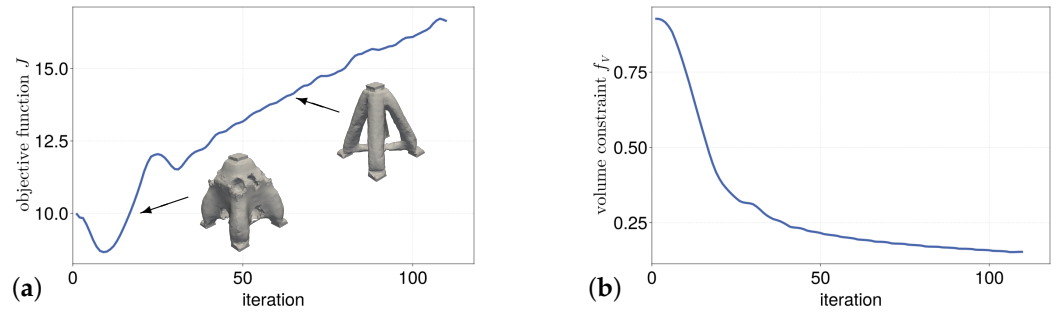


Figure 14. (a) Objective function for the single-material case of the box. (b) Volume constraint for the single-material case of the box.

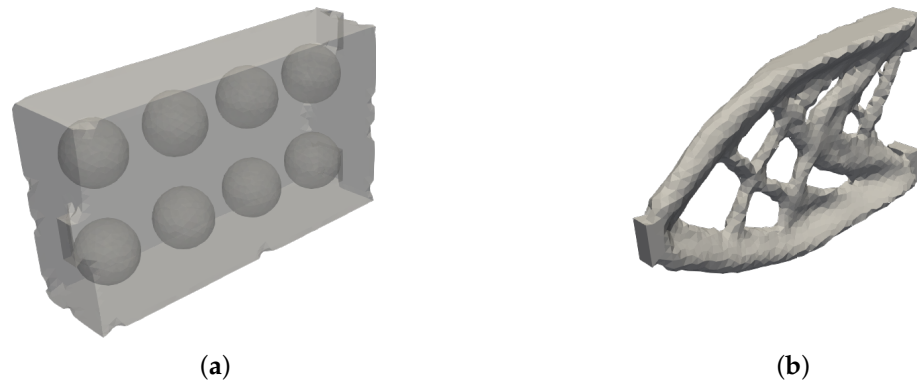


Figure 15. (a) Initial hole structure of the cantilever in the single-material case. (b) Optimization result for the cantilever at iteration 60.

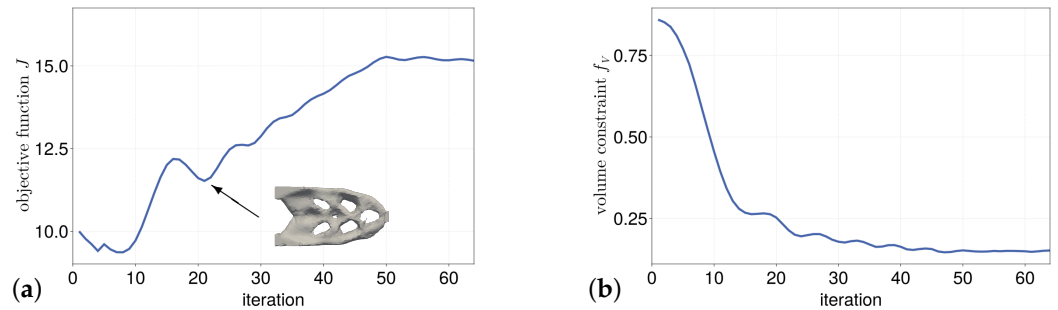


Figure 16. (a) History of the objective function for the single-material case of the cantilever. (b) History of the volume constraint for the single-material case of the cantilever.

In summary, it can be said that the developed optimization method provides plausible results in the single-material case, which are also consistent with the results from the current state of research. As a result, the developed method is considered validated for the single-material case.

3.3. Multi-Material

Two isotropic materials with a Young’s modulus of $E_1 = 70,000$ and $E_2 = 35,000$ are used for the multi-material cases. The Poisson’s ratio for the two materials is $\nu_1 = \nu_2 = 0.3$. The weighting is selected as $w_c = 0.8$, $w_{ri} = 0.15$ and $w_{rs} = 0.05$ for both cases. The two regularization terms are activated after iteration 30 to ensure that they do not dominate the optimization at the beginning. Since the SDF property disappears when the MBO operator is applied, the two level set functions are redistanced in each iteration. The c parameter of the MMA algorithm is increased from a starting value of 0.1 to 400.0 within 250 iterations by means of a linear ramp.

In Figure 17a,b, the initial hole structure for the multi-material case of the box as well as the optimization result is shown. The structure of the multi-material result is very similar to the single-material case. In this case, the square cross strut structure has formed. The less stiff material is shifted by the optimization near the load application.

Figure 18a,b shows the evolution of the objective function and the evolution of the two volume constraints.

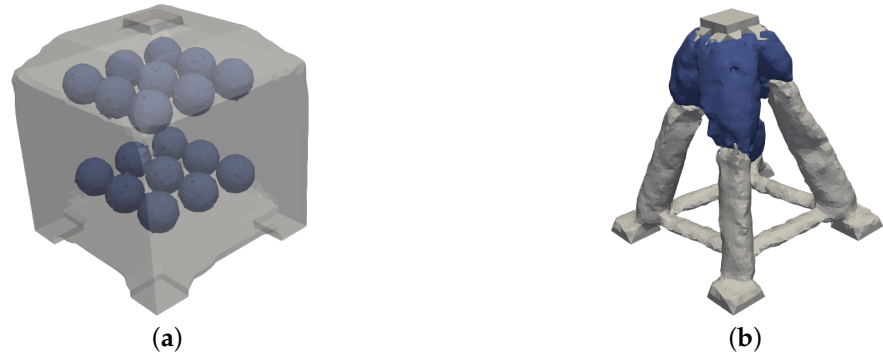


Figure 17. (a) Initial hole structure of the box in the multi-material case. Material 1 fills the entire design space except for the holes that are filled with material 2 (blue). (b) Optimization result for the box at iteration 293.

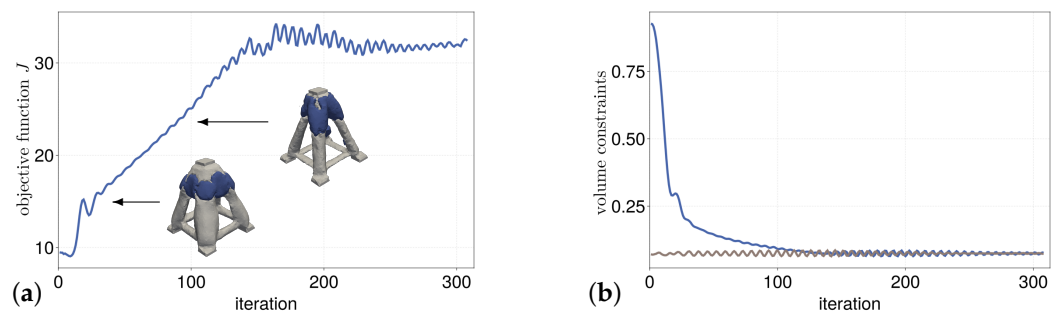


Figure 18. (a) History of the objective function for the multi-material case of the box. (b) History of the volume constraints for the multi-material case of the box (blue material 1, brown material 2).

In Figure 19a,b, the initial hole structures for the multi-material case of the cantilever and the resulting optimization result are shown. The optimization result is very similar to the single-material result. The material with the lower stiffness is shifted to the rear part of the cantilever in particular. It is also clearly visible that the bars have a greater thickness in the areas of the less stiff material (blue) in order to compensate for the disadvantage of the lower Young's modulus.

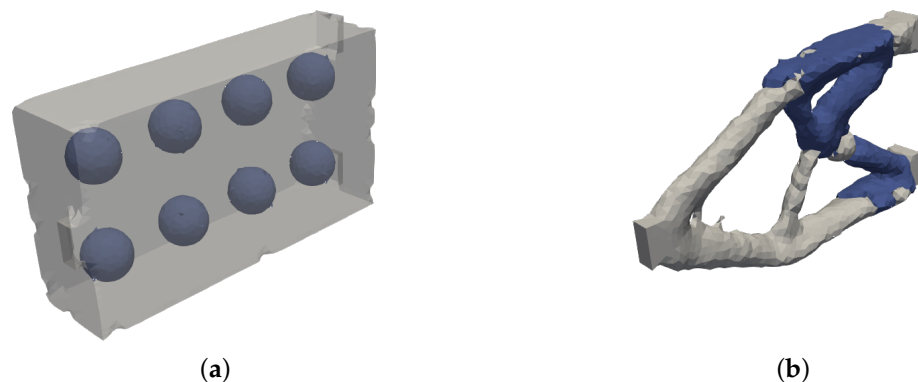


Figure 19. (a) Initial hole structure of the cantilever in the multi-material case. Material 1 fills the entire design space except for the holes that are filled with material 2 (blue). (b) Optimization result of the cantilever at iteration 267.

Figure 20a,b show the evolution of the objective function and the evolution of the two volume-boundary conditions.

Since the developed optimization method for the multi-material case reuses the already validated components from the single-material case and the optimization results for the multi-material case are also plausible, the developed method is also considered validated for the multi-material case.

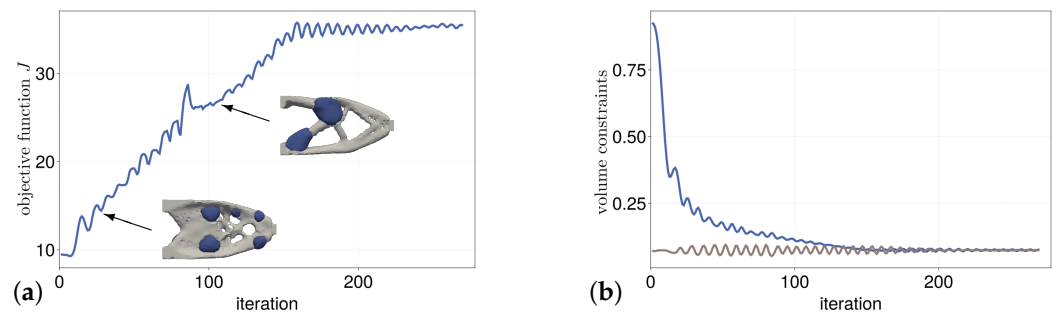


Figure 20. (a) History of the objective function for the multi-material case of the cantilever. (b) History of the volume constraints for the multi-material case of the cantilever (blue material 1, brown material 2).

4. Sensitivity Studies

In the following, the influence of the two parameters, i.e., the element edge length ratio of the two meshes and the filter radius, on the optimization result is examined. One study is carried out for each parameter. In both cases, the symmetrical cantilever clamped on one side serves as the object of investigation for the studies. The values of the other parameters are selected analogously as above.

4.1. Edge Length Ratio

In Figure 21, the optimization results are shown for different edge lengths of the FE mesh with a constant edge length of the level set mesh (change in ratio). It can be seen that the optimization runs in different local optima, but no systematic pattern can be identified. In addition, the local optima are very close to each other in terms of structure and material distribution.

Further numerical experiments have shown that satisfactory results can always be achieved with approximately the same edge length of the two meshes.

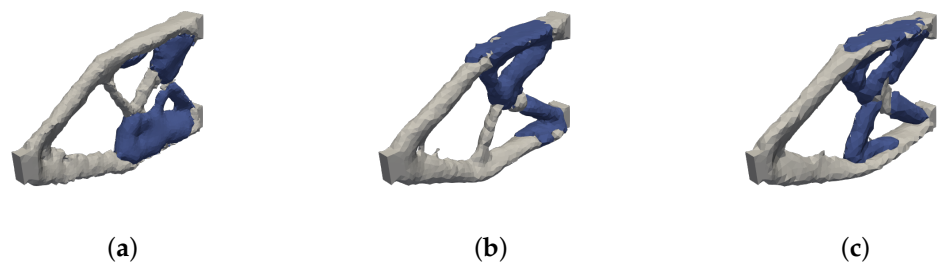


Figure 21. Optimization results of the cantilever for the multi-material case for different average edge lengths of the FE mesh with constant edge length of the LS mesh (a) $length_{avg} = 1.5$ (iteration 230). (b) $length_{avg} = 2.0$ (iteration 267). (c) $length_{avg} = 2.5$ (iteration 163).

4.2. Filter Radius

In Figure 22, the optimization results for three different filter parameters of the Helmholtz filter are listed. It can be seen that this has almost no influence on the final optimization result in the selected range. If the filter radius is increased further, it appears that the optimizer generates coarser features, but the two materials are separated during the optimization, and it is therefore aborted.

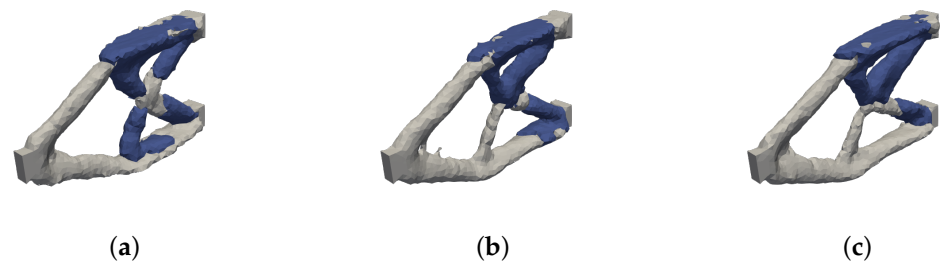


Figure 22. Optimization results of the cantilever for the multi-material case for different values of the Helmholtz filter parameter (a) $r_f = 0.75 * length_{avg}$ (iteration 227). (b) $r_f = 1.25 * length_{avg}$ (iteration 267). (c) $r_f = 1.75 * length_{avg}$ (iteration 205).

5. Conclusions

In this article, a multi-material topology optimization method was developed to support the product developer for the development of structures in multi-material design. The method combines the Velocity Field Level set method with the Reconciled Level set method. The method uses a constant mesh for the development of the level set functions and a second adaptive mesh for the evaluation of the system response. For the validation of the method, it was first validated on two single-material examples and then on two multi-material examples.

The influence of the edge length ratio of the two meshes and the influence of the filter parameter on the optimization result were examined in two sensitivity studies. It was shown that the influence is low as long as the parameters are within a reasonable range.

The method forms the basis for further optimization problems that require an explicit resolution of the structures in the multi-material design during optimization. This includes the authors' current research, in which the influence of the interfaces between the materials, as presented for the first time in [33], is taken into account. The authors plan to publish their findings in the near future.

Author Contributions: Conceptualization, R.R. and A.A.; methodology, R.R.; software, R.R.; validation, R.R. and A.A.; formal analysis, R.R.; investigation, R.R.; data curation, R.R.; writing—original draft preparation, R.R.; writing—review and editing, R.R.; visualization, R.R.; supervision, A.A.; funding acquisition, A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data that support the findings of this manuscript are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FE	Finite Element
FEM	Finite Element Method
LS	Level Set
MDPI	Multidisciplinary Digital Publishing Institute
MBO	Merriman–Bence–Osher
MMA	Method of Moving Asymptotes
RCLSM	Reconciled Level Set Method
VFLSM	Velocity Field Level Set Method

References

1. Li, D.; Kim, I.Y. Multi-material topology optimization for practical lightweight design. *Struct. Multidiscip. Optim.* **2018**, *58*, 1081–1094. [[CrossRef](#)]
2. Feppon, F. Shape and Topology Optimization Applied to Compact Heat Exchangers. 2021. Available online: <https://hal.science/hal-03207863/document> (accessed on 16 August 2024).
3. Borrvall, T.; Petersson, J. Topology optimization of fluids in Stokes flow. *Int. J. Numer. Methods Fluids* **2003**, *41*, 77–107. [[CrossRef](#)]
4. Fraldi, M.; Esposito, L.; Perrella, G.; Cutolo, A.; Cowin, S.C. Topological optimization in hip prosthesis design. *Biomech. Model. Mechanobiol.* **2010**, *9*, 389–402. [[CrossRef](#)]
5. Bendsoe, M.P.; Kikuchi, N. Generating optimal topologies in structural design using a homogenization method. *Comput. Methods Appl. Mech. Eng.* **1988**, *71*, 197–224. [[CrossRef](#)]
6. Bendsoe, M.P. Optimal shape design as a material distribution problem. *Struct. Optim.* **1989**, *1*, 193–202. [[CrossRef](#)]
7. Rozvany, G.I.N. *Structural Design via Optimality Criteria*; Mechanics of Elastic and Inelastic Solids; Springer: Dordrecht, The Netherlands, 1989; Volume 8. [[CrossRef](#)]
8. Wang, M.Y.; Zhou, S. Phase field: A variational method for structural topology optimization. *Comput. Model. Eng. Sci.* **2004**, *6*, 547–566. [[CrossRef](#)]
9. Xie, Y.; Steven, G. A simple evolutionary procedure for structural optimization. *Comput. Struct.* **1993**, *49*, 885–896. [[CrossRef](#)]
10. Querin, O.; Steven, G.; Xie, Y. Evolutionary structural optimisation (ESO) using a bidirectional algorithm. *Eng. Comput.* **1998**, *15*, 1031–1048. [[CrossRef](#)]
11. Sethian, J.; Wiegmann, A. Structural boundary design via level set and immersed interface methods. *J. Comput. Phys.* **2000**, *163*, 489–528. [[CrossRef](#)]
12. Wang, M.Y.; Wang, X.; Guo, D. A level set method for structural topology optimization. *Comput. Methods Appl. Mech. Eng.* **2003**, *192*, 227–246. [[CrossRef](#)]
13. Allaire, G.; Jouve, F.; Toader, A.M. Structural optimization using sensitivity analysis and a level-set method. *J. Comput. Phys.* **2004**, *194*, 363–393. [[CrossRef](#)]
14. Van Dijk, N.P.; Maute, K.; Langelaar, M.; Van Keulen, F. Level-set methods for structural topology optimization: A review. *Struct. Multidiscip. Optim.* **2013**, *48*, 437–472. [[CrossRef](#)]
15. Yaji, K.; Otomori, M.; Yamada, T.; Izui, K.; Nishiwaki, S.; Pironneau, O. Shape and topology optimization based on the convected level set method. *Struct. Multidiscip. Optim.* **2016**, *54*, 659–672. [[CrossRef](#)]
16. Zheng, J.; Zhu, S.; Soleymani, F. A new efficient parametric level set method based on radial basis function-finite difference for structural topology optimization. *Comput. Struct.* **2024**, *297*, 107364. [[CrossRef](#)]
17. Wang, Y.; Kang, Z. A velocity field level set method for shape and topology optimization. *Int. J. Numer. Methods Eng.* **2018**, *115*, 1315–1336. [[CrossRef](#)]
18. Svanberg, K. The method of moving asymptotes—A new method for structural optimization. *Int. J. Numer. Methods Eng.* **1987**, *24*, 359–373. [[CrossRef](#)]
19. Wang, M.Y.; Wang, X. “Color” level sets: A multi-phase method for structural topology optimization with multiple materials. *Comput. Methods Appl. Mech. Eng.* **2004**, *193*, 469–496. [[CrossRef](#)]
20. Merriman, B.; Bence, J.K.; Osher, S.J. Motion of multiple junctions: A level set approach. *J. Comput. Phys.* **1994**, *112*, 334–363. [[CrossRef](#)]
21. Chen, S.; Gonella, S.; Chen, W.; Liu, W.K. A level set approach for optimal design of smart energy harvesters. *Comput. Methods Appl. Mech. Eng.* **2010**, *199*, 2532–2543. [[CrossRef](#)]
22. Vogiatzis, P.; Chen, S.; Wang, X.; Li, T.; Wang, L. Topology optimization of multi-material negative Poisson’s ratio metamaterials using a reconciled level set method. *Comput.-Aided Des.* **2017**, *83*, 15–32. [[CrossRef](#)]
23. Tian, J.; Li, M.; Han, Z.; Chen, Y.; Gu, X.D.; Ge, Q.; Chen, S. Conformal topology optimization of multi-material ferromagnetic soft active structures using an extended level set method. *Comput. Methods Appl. Mech. Eng.* **2022**, *389*, 114394. [[CrossRef](#)]
24. Belytschko, T.; Xiao, S.P.; Parimi, C. Topology optimization with implicit functions and regularization. *Int. J. Numer. Methods Eng.* **2003**, *57*, 1177–1196. [[CrossRef](#)]
25. Burman, E.; Elfverson, D.; Hansbo, P.; Larson, M.G.; Larsson, K. Shape optimization using the cut finite element method. *Comput. Methods Appl. Mech. Eng.* **2018**, *328*, 242–261. [[CrossRef](#)]
26. Ha, S.H.; Cho, S. Level set based topological shape optimization of geometrically nonlinear structures using unstructured mesh. *Comput. Struct.* **2008**, *86*, 1447–1455. [[CrossRef](#)]
27. Allaire, G.; Dapogny, C.; Frey, P. A mesh evolution algorithm based on the level set method for geometry and topology optimization. *Struct. Multidiscip. Optim.* **2013**, *48*, 711–715. [[CrossRef](#)]
28. Li, H.; Yamada, T.; Jolivet, P.; Furuta, K.; Kondoh, T.; Izui, K.; Nishiwaki, S. Full-scale 3D structural topology optimization using adaptive mesh refinement based on the level-set method. *Finite Elem. Anal. Des.* **2021**, *194*, 103561. [[CrossRef](#)]
29. Yamada, T.; Izui, K.; Nishiwaki, S.; Takezawa, A. A topology optimization method based on the level set method incorporating a fictitious interface energy. *Comput. Methods Appl. Mech. Eng.* **2010**, *199*, 2876–2891. [[CrossRef](#)]
30. Osher, S.; Fedkiw, R. *Level Set Methods and Dynamic Implicit Surfaces*; Number 153 in Applied Mathematical Sciences; Springer: New York, NY, USA; Berlin/Heidelberg, Germany, 2003.

31. Aage, N.; Lazarov, B.S. Parallel framework for topology optimization using the method of moving asymptotes. *Struct. Multidiscip. Optim.* **2013**, *47*, 493–505. [[CrossRef](#)]
32. Saye, R. High-order methods for computing distances to implicitly defined surfaces. *Commun. Appl. Math. Comput. Sci.* **2014**, *9*, 107–141. [[CrossRef](#)]
33. Renz, R.; Frank, N.; Albers, A. Three-dimensional multi-material topology optimization considering interface behavior. In Proceedings of the 15th World Congress of Structural and Multidisciplinary Optimization (WCSMO 2023), Cork, UK, 5–9 June 2023.
34. Lazarov, B.S.; Sigmund, O. Filters in topology optimization based on Helmholtz-type differential equations. *Int. J. Numer. Methods Eng.* **2011**, *86*, 765–781. [[CrossRef](#)]
35. Morgan, N.R.; Waltz, J.I. 3D level set methods for evolving fronts on tetrahedral meshes with adaptive mesh refinement. *J. Comput. Phys.* **2017**, *336*, 492–512. [[CrossRef](#)]
36. Bagley, B.; Sastry, S.P.; Whitaker, R.T. A marching-tetrahedra algorithm for feature-preserving meshing of piecewise-smooth implicit surfaces. *Procedia Eng.* **2016**, *163*, 162–174. [[CrossRef](#)]
37. Barill, G.; Dickson, N.G.; Schmidt, R.; Levin, D.I.W.; Jacobson, A. Fast winding numbers for soups and clouds. *ACM Trans. Graph.* **2018**, *37*, 1–12. [[CrossRef](#)]
38. Faraj, N.; Thiery, J.M.; Boubekur, T. Multi-material adaptive volume remesher. *Comput. Graph.* **2016**, *58*, 150–160. [[CrossRef](#)]
39. Parthasarathy, V.; Graichen, C.; Hathaway, A. A comparison of tetrahedron quality measures. *Finite Elem. Anal. Des.* **1994**, *15*, 255–261. [[CrossRef](#)]
40. Dassi, F.; Formaggia, L.; Zonca, S. Degenerate tetrahedra removal. *Appl. Numer. Math.* **2016**, *110*, 1–13. [[CrossRef](#)]
41. Loseille, A.; Menier, V. Serial and parallel mesh modification through a unique cavity-based primitive. In Proceedings of the 22nd International Meshing Roundtable, Orlando, FL, USA, 13–16 October 2013; Sarrate, J., Staten, M., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 541–558. [[CrossRef](#)]
42. Devillers, O.; Pion, S. *Efficient Exact Geometric Predicates for Delaunay Triangulations*; Technical Report RR-4351; INRIA: Sophia Antipolis, France, 2002.
43. Shang, M.; Zhu, C.; Chen, J.; Xiao, Z.; Zheng, Y. A parallel local reconnection approach for tetrahedral mesh improvement. *Procedia Eng.* **2016**, *163*, 289–301. [[CrossRef](#)]
44. Holoch, J.; Lenhardt, S.; Renz, R.; Albers, A. Investigation on the influence of different modeling of multiple surface layers on a 3d topology optimization. In Proceedings of the NAFEMS World Congress, Online, 25–29 October 2021.
45. Hu, Y.; Schneider, T.; Wang, B.; Zorin, D.; Panozzo, D. Fast tetrahedral meshing in the wild. *ACM Trans. Graph.* **2020**, *39*, 117:1–117:18. [[CrossRef](#)]
46. Henrot, A.; Pierre, M. *Shape Variation and Optimization: A Geometrical Analysis*, 1st ed.; EMS Tracts in Mathematics; EMS Press: Helsinki, Finland, 2018; Volume 28. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.