Structural and Multidisciplinary Optimization (2025) 68:112
https://doi.org/10.1007/s00158-025-04026-7

RESEARCH PAPER

On choosing interpolation domains and functions in topology optimization of multi-scale structures: a comparative study

Ahmed Mohamed Jubartalla Ali[1,2] · **Abdulmajeed Altassan**[3] · **Théodore Cherrière**[4] · **Peter Gangl**[4] · **Margit Gföhler**[2] · **Mario Kapl**[5]

Received: 9 January 2025 / Revised: 19 March 2025 / Accepted: 25 April 2025 / Published online: 14 June 2025

Abstract

In this work, we evaluate the behavior of several material interpolation schemes in conducting topology optimization of composite structures made of 28 orthotropic materials with similar costs and densities. The materials are categorized into three families according to their geometrical features and arranged in a collection of interpolation domains, including 2D and 3D domains, as well as orthogonal and non-orthogonal ones. The concept of Coons patches is used to extend three interpolation functions, namely polynomial fitting, ordered Solid Isotropic Material with Penalization (ordered SIMP), and smooth single-variable-based interpolation, from 1D to 2D and 3D. Several material interpolation schemes are built using different combinations of interpolation domains and interpolation functions for a representative comparison. Numerical experiments are conducted using a two-scale topology optimization procedure to solve a benchmark problem given by the Messerschmitt-Bölkow-Blohm (MBB) beam problem. The obtained results show that, when taking the simple case of compliance minimization, as well as candidate lattices of same densities, the selection of the interpolation domain and of the interpolation function does not have a significant effect on the optimization process. Accordingly, we conclude that taking the simplest combination of a 2D square domain together with a piecewise linear function is sufficient to get the optimized design, if the candidate lattices are well categorized and sorted.

Keywords Topology optimization · Multi-scale structure · Interpolation domain · Interpolation function · Coons patches

1 Introduction

Due to the high manufacturing flexibility offered by Additive Manufacturing, Multi-Material Topology Optimization (MMTO) of lattice structures has received considerable momentum in recent research. The main challenge in MMTO is to perform it without significantly increasing the cost of renewing the design variables, which is mainly related to the number of used materials and can be controlled using proper material interpolation.

In the density-based approach, also known as the SIMP (Solid Isotropic Material with Penalization) method (Bendsøe 1989; Zhou and Rozvany 1991), Topology Optimization (TO) is conducted by first placing the two candidate materials (void and solid) over the vertices of a line interpolation domain ($\rho \in [0, 1]$), typically void (white) at $\rho = 0$ and solid (black) at $\rho = 1$. In this paper, the term "interpolation domain" refers to the domain of definition of the interpolation function. To avert solving such a 0–1 binary optimization problem, a non-physical transitory phase (gray) between

Responsible editor: Xu Guo.

✉ Ahmed Mohamed Jubartalla Ali
a.ali@fh-kaernten.at

[1] ADMiRE Research Center, Carinthia University of Applied Sciences, Europastraße 4, 9524 Villach, Austria

[2] Research Unit of Biomechanics and Rehabilitation Engineering, TU Wien, Lehargasse 6, 1060 Vienna, Austria

[3] George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, 801 Ferst Dr., Atlanta, GA 30332, USA

[4] Johann Radon Institute for Computational and Applied Mathematics (RICAM), Altenberger Straße 69, 4040 Linz, Austria

[5] Department of Engineering & IT, Carinthia University of Applied Sciences, Europastraße 4, 9524 Villach, Austria

void and solid is introduced, which should also be made less effective by use of a penalized interpolation of the elements stiffness such as the modified power-law (Andreassen et al. 2010)

$$\mathbf{E}(\boldsymbol{\rho}) = (1-\boldsymbol{\rho}^p)\,E_{\text{void}} + \boldsymbol{\rho}^p E_{\text{solid}}. \tag{1}$$

Here, $\boldsymbol{\rho} = \{\boldsymbol{\rho}_1, \boldsymbol{\rho}_2, \boldsymbol{\rho}_3, \ldots, \boldsymbol{\rho}_N\}^T$ is a vector that contains the pseudo densities of the N finite elements; E_{solid} is the Young's modulus for the used material; E_{void} is a small positive number to avoid singularity; and $p > 1$ is a penalization factor. Equation (1) might also be solved without the relaxation-penalization scheme as in Sivapuram and Picelli (2017), Liang and Cheng (2019) and Liang et al. (2020).

In MMTO, the SIMP method could be extended to account for three or four materials by replacing the 1D line interpolation domain by 2D square or 3D cube interpolation domains, respectively, and similarly placing the candidate materials over the corresponding vertices as illustrated in Fig. 1(top). Then, the power-laws for interpolating three and four materials read (Bendsøe and Sigmund 2004)

$$\begin{aligned}\mathbf{E}(\boldsymbol{\rho}_1,\boldsymbol{\rho}_2) &= (1-\boldsymbol{\rho}_1^p)\,E_{\text{void}} \\ &\quad + \boldsymbol{\rho}_1^p\big((1-\boldsymbol{\rho}_2^p)\,E_1 + \boldsymbol{\rho}_2^p E_2\big), \quad \text{and} \\ \mathbf{E}(\boldsymbol{\rho}_1,\boldsymbol{\rho}_2,\boldsymbol{\rho}_3) &= (1-\boldsymbol{\rho}_1^p)\,E_{\text{void}} + \boldsymbol{\rho}_1^p\big((1-\boldsymbol{\rho}_2^p)\,E_1 \\ &\quad + \boldsymbol{\rho}_2^p\big((1-\boldsymbol{\rho}_3^p)\,E_2 + \boldsymbol{\rho}_4^p E_3\big)\big),\end{aligned} \tag{2}$$

respectively, where $\boldsymbol{\rho}_1, \boldsymbol{\rho}_2, \boldsymbol{\rho}_3$ are the three design variables; and E_1, E_2, E_3 are the Young's modulus for the first, second, and third materials, respectively. Further extensions to include N_M materials is theoretically possible by using an $(N_M - 1)$-D hypercube interpolation domain. However, the usage of the extended SIMP method in the literature is limited to three and four candidate materials (including void) (Gibiansky and Sigmund 2000; Taheri and Suresh 2016; Gaynor et al. 2014; Gao and Zhang 2011), as a study by Stegmann and Lund (2005) showed that including more than four candidate materials tends to converge to bad local minima.

An alternative method that is capable for handling higher numbers of candidate materials is the Discrete Material Optimization (DMO) (Stegmann and Lund

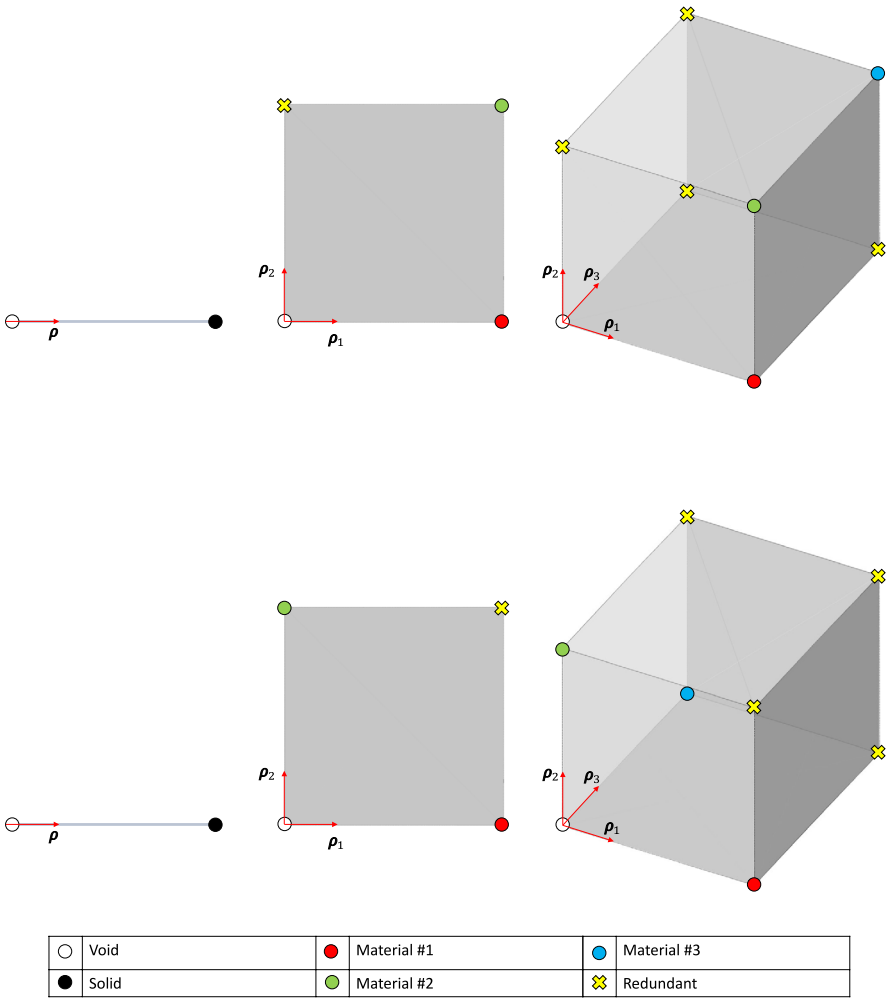


Fig. 1 Orthogonal material interpolation domains used in the SIMP (top) and DMO (bottom) interpolation schemes for accommodating two (left), three (middle), and four (right) candidate materials

2005), which illustrated capabilities for optimizing the orientation of composite laminate based on an orthotropic material with 12 different angles (Stegmann and Lund 2005; Lund and Stegmann 2004), as well as material selection problems with 6 (Sanders et al. 2018a) and up to 15 (Sanders et al. 2018b) candidate materials. This method expresses the element elasticity tensor $\mathbf{D}_e$ as a weighted sum of the elasticity tensors of all candidate materials (Stegmann and Lund 2005). Similarly, the vector contains the isotropic properties for the N mesh elements interpolated as

$$\mathbf{E} = \sum_{i=1}^{N_M-1} \boldsymbol{\omega}_i E_i = \boldsymbol{\omega}_1 E_1 + \boldsymbol{\omega}_2 E_2 + \boldsymbol{\omega}_3 E_3 + \cdots + \boldsymbol{\omega}_{N_M-1} E_{N_M-1}, \tag{3}$$

where $\boldsymbol{\omega}_i \in [0, 1]$ are the weights defined as

$$\boldsymbol{\omega}_i = \boldsymbol{\rho}_i^p \prod_{j=1}^{N_M-1} \left(1 - \boldsymbol{\rho}_{j\neq i}^p\right).$$

As seen in Eq. (3), the DMO method, unlike the extended SIMP method (Eq. (2)), does not explicitly consider void as a candidate material, as zero-stiffness appears when all design variables are zero or when more than one design variable equals 1. However, the two methods require using an $(N_M - 1)$-D orthogonal material interpolation domain, but with a slightly different material placement as compared in Fig. 1.

It is not that simple to determine whether the DMO method or extended SIMP method is better than the other. For instance, a study done by Gao and Zhang (2011) showed that DMO converges to better local minima than extended SIMP, even when only three candidate materials are used. On the other hand, the study of Liu et al. (2023) demonstrated that a discrete formulation of the extended SIMP method can effectively solve MMTO problems as Eq. (2) has separable characteristics in sensitivity analysis of discrete variables in comparison to Eq. (3).

Furthermore, the study of Yi et al. (2023), proposes a unified material interpolation method that follows the DMO material placement [Fig. 1(bottom)], but with an alternative mapping based on the p-norm of the design variables in order to model each material equally. The method showed clear convergence to 0–1 solution for each material in comparison to both DMO and extended SIMP. Similarly, Huang and Li (2021) obtained clear 0–1 designs using an interpolation method based on the physical volume fractions of candidate materials within each element.

To sum up, the most common thing between the extended SIMP, DMO methods, unified material interpolation, and other multivariate methods, is the need for $N_M - 1$ design variables per mesh element e. This makes the memory cost of renewing the total $N_M \times N_E$ design variables higher with increasing the number of candidate materials, especially when optimizing Functionally Graded Lattices (FGL) where the number of candidate materials (lattices) easily exceeds ten.

This led to another approach that decouples the dimensionality of the interpolation domains from the number of candidate materials. For instance, the Shape Functions with Penalization (SPF) scheme proposed by Bruyneel (2011) employs the conventional bi-linear shape function of the first-order quadrangular finite element for interpolating between four candidate materials placed over the vertices of a 2D square domain, which resulted in using only two design variables instead of four. According to the author, this could also be extended to include up to seven candidate materials and stay in 2D space by using shape functions with mixed degrees on the edges. However, for including more than seven candidate materials, considering a 3D cube domain would be recommended. Figure 2 shows the interpolation domain and material placement used with SFP.

The dimensionality of the interpolation domain could be further reduced by using univariate interpolation methods which use only one design variable per mesh element regardless of the number of used materials. In this approach, the N_M candidate materials are placed over a 1D line interpolation domain ($\rho \in [0, 1]$), according to their normalized densities $\boldsymbol{\rho}_1, \boldsymbol{\rho}_2, \boldsymbol{\rho}_3, \ldots, \boldsymbol{\rho}_{N_M-1}$, and then the properties between each two adjacent materials, for instance E_{i-1} and E_i, are interpolated by a continuous function, as illustrated in Fig. 3.

In the literature, a wide variety of continuous functions are used to describe the transitions between the candidate materials (i.e., the red, green, and blue lines in

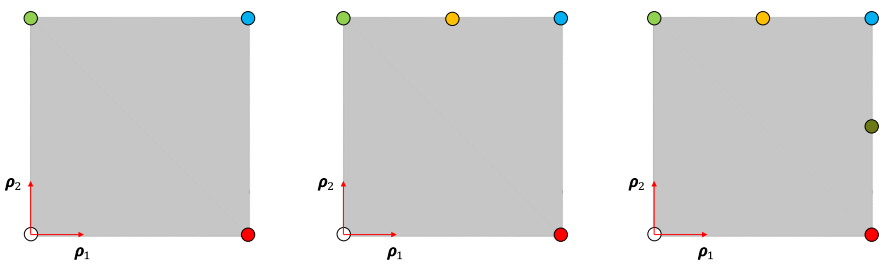


Fig. 2 Square material interpolation domain used in the SFP interpolation scheme for accommodating four candidate materials (left) and possible formulations for accommodating five (middle), and six (right) candidate materials

Fig. 3(right)). For instance, the researchers used a straight line (Liu et al. 2021), SIMP power-law (Zuo and Saitou 2016; Augusto and Palma 2022; Nguyen and Lee 2024), SIMP-like power-law (Xu et al. 2021), peak function (Yin and Ananthasuresh 2001), stair-form function (Liao et al. 2024), Heaviside function (Liao 2021; Jiang et al. 2024), and smoothened Heaviside function (Dinh et al. 2024). In addition, multi-valued integer programming is also used to solve multi-material optimization problems without the relaxation-penalization approach (Deng et al. 2024).

The candidate materials could also be sorted based on other physical properties, such as Young's modulus. This would be very useful in case of using candidate materials with similar or close densities, as in Guo et al. (2024) and Wan et al. (2024), and when optimizing FGL, where it is practical to decouple between the solid-void phases and the candidate lattices by using two distinct interpolation domains and bridge them by the Porous Anisotropic Material with Penalization (PAMP) method (Liu et al. 2008), as illustrated in Fig. 4. On the one hand, and as presented before, the void and solid phases are typically placed over the vertices of a line domain and interpolated through the SIMP power-law (Eq. (1)). On the other hand, for interpolating between the lattices, the designers have more freedom in choosing the interpolation domain ($\mathcal{D}$) and interpolation function (f). In the two following paragraphs, we discuss them in more details.

How to choose the interpolation domain ($\mathcal{D}$)? In the simple case of using one family of FGL (i.e., a group sharing similar geometrical features), as in Wang et al. (2018, 2020a), employing a line interpolation domain $\boldsymbol{\varphi} \in [0, 1]$ is sufficient. However, when more families (having different densities or different geometrical features, for instance) are considered, the 1D line interpolation domain is typically extended orthogonally (as in Fig. 1) to 2D square (Costa et al. 2022; Wang et al. 2020b, 2023), 3D cube (Zhang et al. 2023; Yang et al. 2024), 4D hypercube (Zhang et al. 2022; Zhou et al. 2022), or higher. To the best of our knowledge, using custom-designed non-orthogonal domains in TO of FGL have not yet been studied. As these domains require additional processes, such as treating the strayed design points (i.e., points goes outside the domain), they also eliminate the existence of redundant entries seen in Fig. 1 as we suggest that the permutation of the families affects the optimization process and may promote bad local minima. Alternatively, motivated by the study of using a diamond

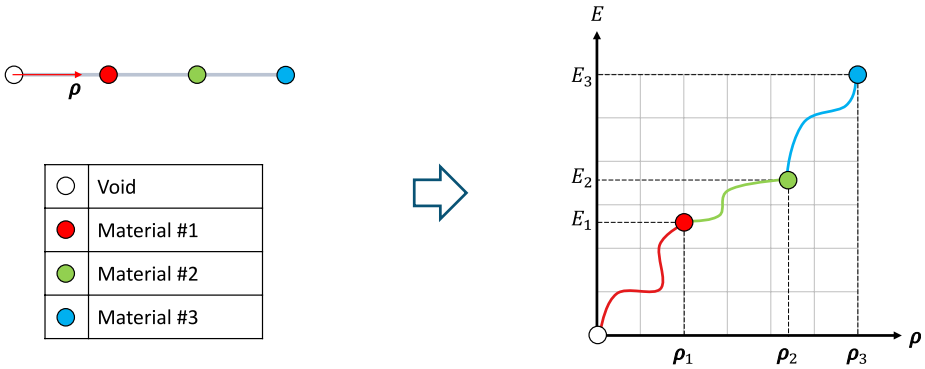


Fig. 3 Four candidate materials sorted over a line interpolation domain according to their densities (left) and the corresponding interpolated property between each two materials (right)

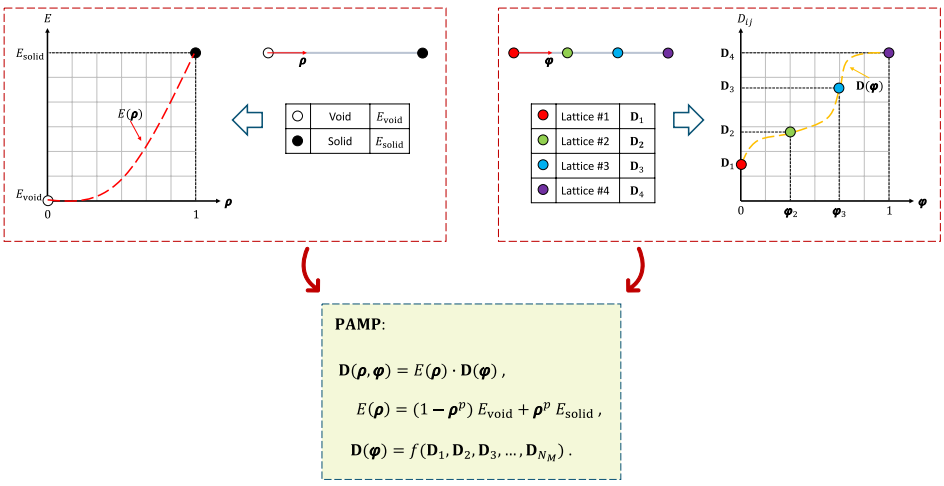


Fig. 4 Two distinct line interpolation domains used for accommodating the solid-void phases (top left) and four lattices (top right), and then bridged by the PAMP method (bottom)

interpolation domain in designing a multiphase electrical machine's stator (Cherrière, Théodore et al. 2022), we propose using equilateral non-orthogonal domains as they provide equal distances between any two families, and more importantly, there is no need to duplicate some families to fill the redundant edges. Examples of accommodating three families of FGL in cube and triangular prism domains are given in Fig. 5.

How to choose the interpolation function (f)? For instance, the polynomial fitting method is very simple and provides continuous derivatives. Depending on the definition of the selection parameter $\boldsymbol{\varphi}$, the fitting degree could be chosen as 2 (Zhang et al. 2022, 2023), 3 (Wang et al. 2018; Yang et al. 2024), or 5 (Wang et al. 2020a, b; Costa et al. 2022). Another alternative method that exhibits local penalization to push $\boldsymbol{\varphi}_e$ more toward the candidate lattices is the ordered SIMP (Zuo and Saitou 2016; Augusto and Palma 2022) and applied in Zhang et al. (2018) and Gu et al. (2022). Furthermore, a recent method with both local penalization and continuous derivatives has been proposed by Dinh et al. (2024) but not yet employed in TO of FGL. In this paper, we study, on the one hand, the effect of using custom-designed material interpolation domains for accommodating three families of FGL. Our compared domains include 2D and 3D, as well as orthogonal and non-orthogonal ones. On the other hand, we also study replacing the classical polynomial fitting method with some advanced methods that offer local penalization and smoothening, namely the ordered SIMP (Zuo and Saitou 2016; Augusto and Palma 2022) and smooth

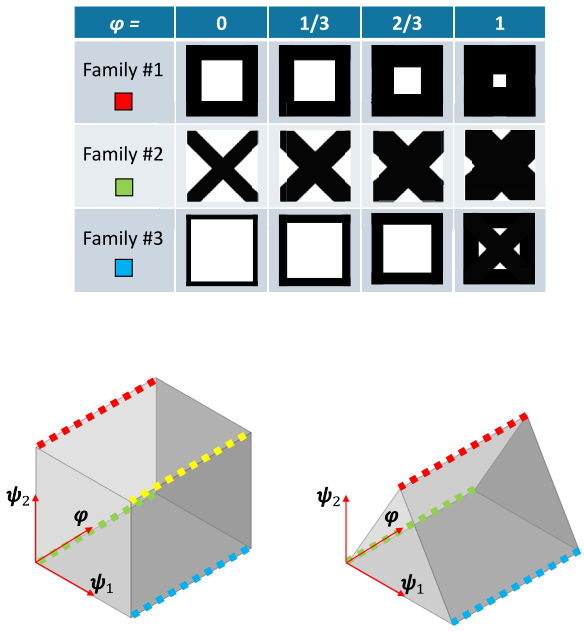


Fig. 5 Three families of FGL (top) accommodated in orthogonal (bottom left) and non-orthogonal (bottom right) interpolation domains

single-variable interpolation (Dinh et al. 2024). One of the novelties of our paper is that we use the concept of Coons patches (Coons 1967) to extend the univariate methods to 2D and 3D domains and use only two and three design variables, respectively, regardless of the number of used lattices.

The rest of this paper is structured as follows. In Sect. 2, we formulate our topology optimization problem and describe the technique used in solving it. Then, the studied interpolation domains and interpolation functions are introduced in Sect. 3. After that, in Sect. 4, the optimization problem and the results are presented and discussed. Finally, we conclude the comparison in Sect. 5.

2 Topology optimization formulation

In this section, we first introduce our materials and design variables. Then, we define our optimization problem and describe the technique used to solve it.

2.1 Materials and their properties

In this study, we use 28 lattices (orthotropic materials), which were developed in a previous work (Ali et al. 2024). These lattices have identical costs and weights and are grouped into three families according to their geometries. For navigating between the lattices, in the previous work, we used the design variables $\boldsymbol{\psi}$ and $\boldsymbol{\varphi}$ to indicate the family (row) and lattice (column), respectively, as shown in Fig. 6(top). In this study, we investigate alternative ways as follows in Sect. 3.1.

The stiffness of each mesh element e is calculated, using the Porous Anisotropic Material with Penalization (PAMP) method (Liu et al. 2008), as

$$\mathbf{D}_e(\boldsymbol{\rho}_e, \boldsymbol{\psi}_e, \boldsymbol{\varphi}_e) = E_e^{\text{den}}(\boldsymbol{\rho}_e)\, \mathbf{D}_e^{\text{mat}}(\boldsymbol{\psi}_e, \boldsymbol{\varphi}_e), \tag{4}$$

where ρ_e is the pseudo density used to distinguish between solid and void. The first component E_e^{den} depends on the pseudo density $\boldsymbol{\rho}_e$ and is obtained by using the modified SIMP method (Sigmund 2007) via

$$E_e^{\text{den}} = E_{\text{min}} + \boldsymbol{\rho}_e^p (E_{\text{o}} - E_{\text{min}}), \tag{5}$$

where p is a penalty factor; E_{o} is Young's modulus of the isotropic material; E_{min} is a very small positive number to prevent the singularity of the stiffness matrix. Here, we typically use $p = 3$, $E_{\text{o}} = 1$, and $E_{\text{min}} = 10^{-6}$.

The second component $\mathbf{D}_e^{\text{mat}}$ depends on the material and is evaluated through numerical homogenization (Andreassen and Andreasen 2014)

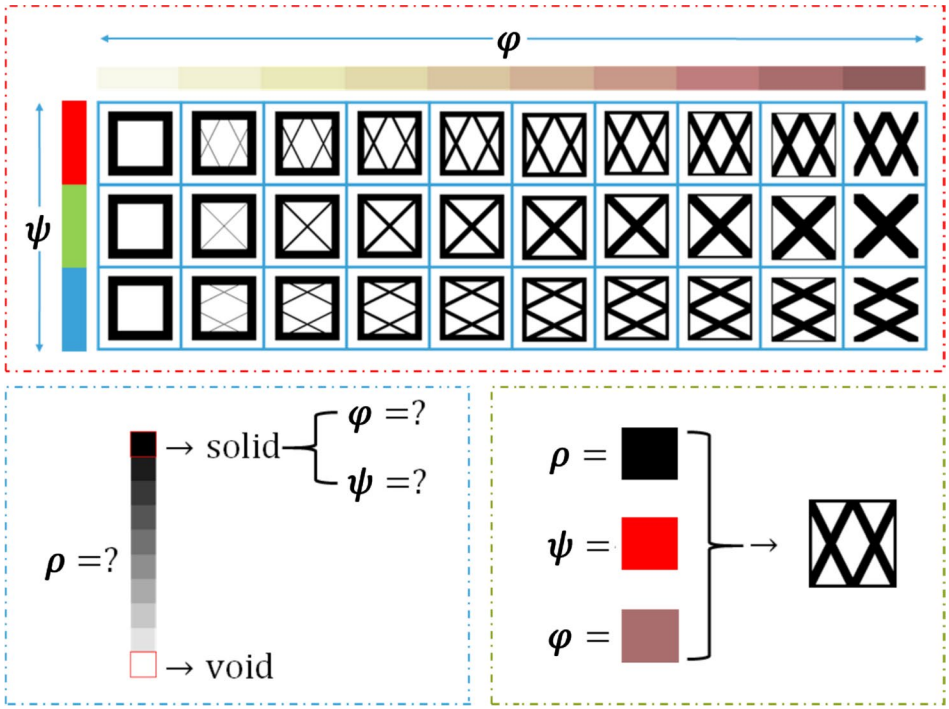


Fig. 6 Lattices grouped in three families (top), design variables (bottom left), and an illustration for selecting the material (bottom right)

$$(\mathbf{D}_e^{\text{mat}})_{ij} = \frac{1}{|\Omega_e|} \sum_{\mu=1}^{M} \left(\mathbf{u}_\mu^{(i)}\right)^T \mathbf{k}_\mu(\rho_\mu^{\text{MI}}) \mathbf{u}_\mu^{(j)}, \tag{6}$$

where $(\mathbf{D}_e^{\text{mat}})_{ij}$ is the (i, j)-th component of the homogenized elasticity tensor for $i, j \in \{1, 2, 6\}$; Ω_e and $|\Omega_e|$ are the domain and volume of the element e, respectively; M is the total number of micro-elements; $\mathbf{u}_\mu^{(i)}$ is the element displacement solution corresponding to the unit strain field $\epsilon^{0(i)}$ which is $(1, 0, 0)^T$, $(0, 1, 0)^T$, and $(0, 0, 1)^T$ for $i = 1, 2$, and 6, respectively; The superscript MI indicates the micro-scale and $\mathbf{k}_\mu$ is the stiffness matrix for the micro-element μ, calculated as

$$\mathbf{k}_\mu = \int_{\Omega_\mu} \mathbf{B}^T \mathbf{D}_\mu \mathbf{B} \, \mathrm{d}\Omega_\mu,$$

with $\mathbf{B}$ as the strain–displacement matrix; and

$$\mathbf{D}_\mu = \frac{E_{\text{o}}^{\text{MI}}}{(1+\nu)(\alpha-\nu)} \begin{bmatrix} \alpha & \nu & 0 \\ \nu & \alpha & 0 \\ 0 & 0 & \frac{\alpha-\nu}{2} \end{bmatrix},$$

and

$$\alpha = \begin{cases} 1, & \text{plane-stress} \\ 1-\nu, & \text{plane-strain} \end{cases}$$

where E_{o}^{MI} and ν are the Young's modulus and Poisson's ratio for the base material. Here, we assume plane-stress state and choose $M = 114$, which captures well the geometrical features of our micro-structures and satisfies the assumption of length-scale separation, i.e., the micro-structures are much smaller than the macro-structures. We also take $E_{\text{o}}^{\text{MI}} = 1$ and $\nu = 0.3$. The obtained equivalent moduli of the lattices are plotted in Fig. 7.

2.2 Optimization problem

Since the density and cost of our materials are the same, they will not be considered in our density-based topology optimization formulation. With the objective of minimizing structural compliance and constraining the volume, our optimization problem can be formulated as follows: Minimize

$$C(\boldsymbol{\rho}, \boldsymbol{\psi}, \boldsymbol{\varphi}) = \mathbf{U}^T(\boldsymbol{\rho}, \boldsymbol{\psi}, \boldsymbol{\varphi}) \, \mathbf{K}(\boldsymbol{\rho}, \boldsymbol{\psi}, \boldsymbol{\varphi}) \, \mathbf{U}(\boldsymbol{\rho}, \boldsymbol{\psi}, \boldsymbol{\varphi}) \tag{7}$$

such that

$$\mathbf{K}(\boldsymbol{\rho}, \boldsymbol{\psi}, \boldsymbol{\varphi}) \, \mathbf{U}(\boldsymbol{\rho}, \boldsymbol{\psi}, \boldsymbol{\varphi}) = \mathbf{F},$$

$$G_1(\boldsymbol{\rho}) = \sum_{e=1}^{N} \boldsymbol{\rho}_e - N \cdot V_{\text{f}} \leq 0,$$

and

$$0 \leq \boldsymbol{\rho}_e \leq 1,$$
$$(\boldsymbol{\psi}_e, \boldsymbol{\varphi}_e) \in \mathcal{D},$$

where C is the structural compliance; $\mathbf{K}$ is the stiffness matrix; $\mathbf{U}$ and $\mathbf{F}$ are the displacement and force vector, respectively; G_1 is the first inequality constraint and

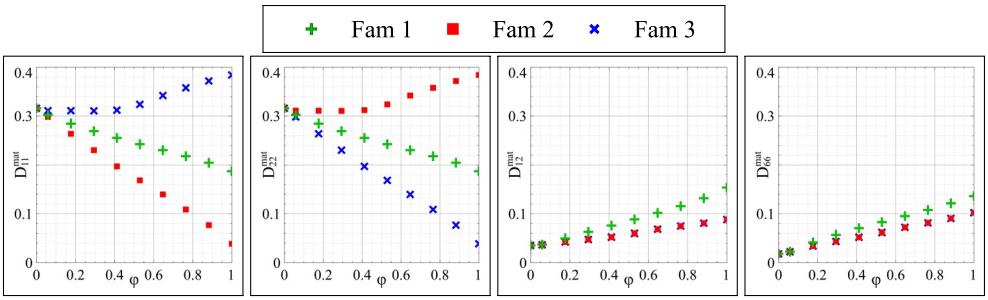


Fig. 7 Effective elastic moduli for the lattices

represents the material volume (G_2 and G_3 are used in Sect. 3.1 and illustrated in Fig. 12); V_f is the maximum allowed volume fraction; and $\mathcal{D}$ is an interpolation domain and should not be confused with the design domain Ω. The former is used to accommodate the candidate materials (Figs. 1, 3, and 5), while the latter represents the allowable space where material can be added or removed during the optimization. It is worth mentioning that the dimensionality of the interpolation domain and design domain are not linked at all. For instance, a 1D interpolation domain could be used in optimizing 2D or 3D structures. Further details about interpolation domains are given in Sect. 3.1.

The sensitivities of the objective and constraint functions were derived in Ali et al. (2024). For the sake of brevity, we directly present the final formulas, which are given as

$$\frac{\partial C}{\partial \boldsymbol{\rho}_e} = -\frac{\partial E_e^{\text{den}}}{\partial \boldsymbol{\rho}_e}\mathbf{u}_e^T\left(\int_{\Omega_e}\mathbf{B}^T\mathbf{D}_e^{\text{mat}}\mathbf{B}\mathrm{d}\Omega_e\right)\mathbf{u}_e, \tag{8}$$

$$\frac{\partial C}{\partial \boldsymbol{\psi}_e} = -E_e^{\text{den}}\mathbf{u}_e^T\left(\int_{\Omega_e}\mathbf{B}^T\frac{\partial\mathbf{D}_e^{\text{mat}}}{\partial \boldsymbol{\psi}_e}\mathbf{B}\mathrm{d}\Omega_e\right)\mathbf{u}_e, \tag{9}$$

$$\begin{gathered}\frac{\partial C}{\partial \boldsymbol{\varphi}_e} = -E_e^{\text{den}}\mathbf{u}_e^T\left(\int_{\Omega_e}\mathbf{B}^T\frac{\partial\mathbf{D}_e^{\text{mat}}}{\partial \boldsymbol{\varphi}_e}\mathbf{B}\mathrm{d}\Omega_e\right)\mathbf{u}_e,\\ \frac{\partial G_1(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}_e} = 1,\end{gathered} \tag{10}$$

where $\mathbf{B}$ is the strain–displacement matrix; and the terms $\partial\mathbf{D}_e^{\text{mat}}/\partial\boldsymbol{\psi}_e$ and $\partial\mathbf{D}_e^{\text{mat}}/\partial\boldsymbol{\varphi}_e$ are the gradients of the interpolating functions presented in Sec. 3.2. The term $\partial E_e^{\text{den}}/\partial\boldsymbol{\rho}_e$ is obtained directly from Eq. (5) via

$$\frac{\partial E_e^{\text{den}}}{\partial \boldsymbol{\rho}_e} = p\boldsymbol{\rho}_e^{p-1}(E_o - E_{\min}). \tag{11}$$

2.3 Optimization algorithm

At the beginning of the optimization process, we set all the design variables to the centroids of the considered interpolation domain, and then update all of them concurrently. The optimization is conducted using the Method of Moving Asymptotes (MMA) (Deetman 2019) and is repeated until the improvement in the compliance is less than 0.1% for ten successive iterations. To prevent the checkerboard problem, $\boldsymbol{\rho}$ is filtered by using the sensitivity filter (Sigmund 1997), while the density filter (Bourdin 2001) is applied for filtering $\boldsymbol{\psi}$ and $\boldsymbol{\varphi}$. In all cases, the minimum filter radius is set to $R_{\min} = 2.0$, i.e., to two mesh elements.

3 Material interpolation schemes

In this study, we concentrate on two aspects of multi-material modeling: which interpolation domains should be used and how the relation between our materials should be described. We explore several possibilities to answer these two questions in the following subsections.

3.1 Interpolation domains

How we place (arrange) our materials depends on the shape of the selected interpolation domain. We denote by $\mathcal{D}$ the interpolation domain, which is a convex polytope containing the admissible interpolation variables ($\boldsymbol{\psi}_e, \boldsymbol{\varphi}_e$). Generally, the candidate materials are placed over nodes inside the domain $\mathcal{D}$ (see Figs. 9, 10, 11, and 13), so that when an interpolation variable is equal to a node coordinate, it designates a "real" candidate material. Otherwise, it is associated with a "gray" intermediate material. In classical void/solid topology optimization, the only possible interpolation domain is a segment with only two vertices, while in MMTO, many interpolation domains are possible. The choice of $\mathcal{D}$ affects the optimization progress and may lead to convergence to a bad local minimum.

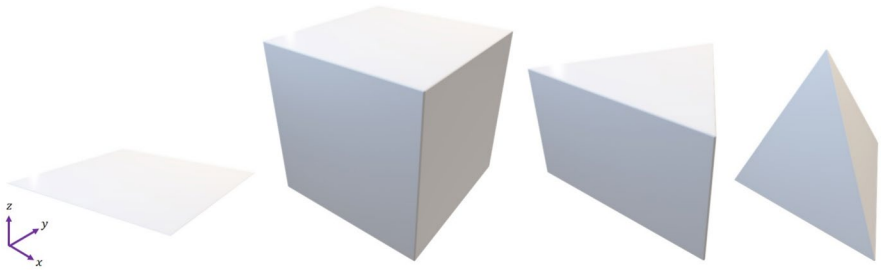


Fig. 8 Interpolation domains (from left to right): Square, cube, triangular prism, and triangular pyramid

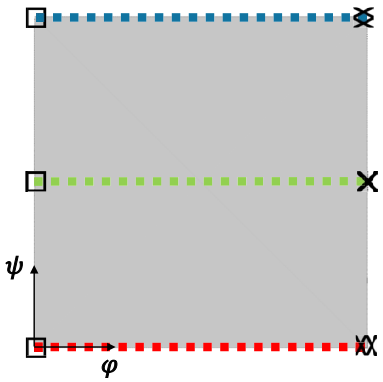
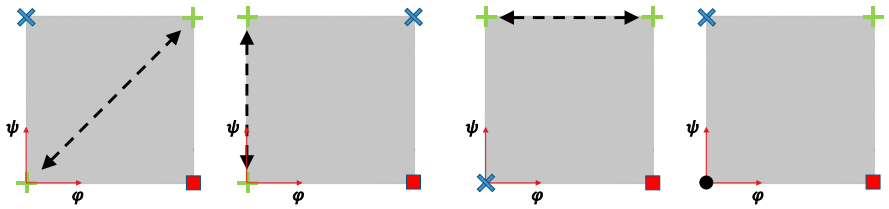


Fig. 9 Square domain: material placement

In this work, we aim to compare 2D and 3D interpolation domains, as well as orthogonal and non-orthogonal domains. The main advantage of using orthogonal domains is the ease of keeping the design points inside the domain (i.e., $(\boldsymbol{\psi}_e, \boldsymbol{\varphi}_e) \in \mathcal{D}$) by means of box constraints with the following projection operator:

$$\mathcal{P} : x_e \mapsto \min(x_{\max}, \max(x_{\min}, x_e)), \tag{12}$$

where $\mathcal{P}$ is the orthogonal projection operator onto an orthogonal domain; x_e is an optimization variable (for instance, $\boldsymbol{\psi}_e$ or $\boldsymbol{\varphi}_e$) lying between two bounds $x_{\min}$ and $x_{\max}$. It is worth mentioning that Eq. (12) is already developed in the MMA solver, and one only needs to define the parameters $x_{\min}$ and $x_{\max}$, according to the bounds of the interpolation domain.



From left to right: Case I, Case II, Case III, and Case IV. Black, green, red, and blue markers represent families #0, #1, #2, and #3, respectively, where family #0 contains voids.

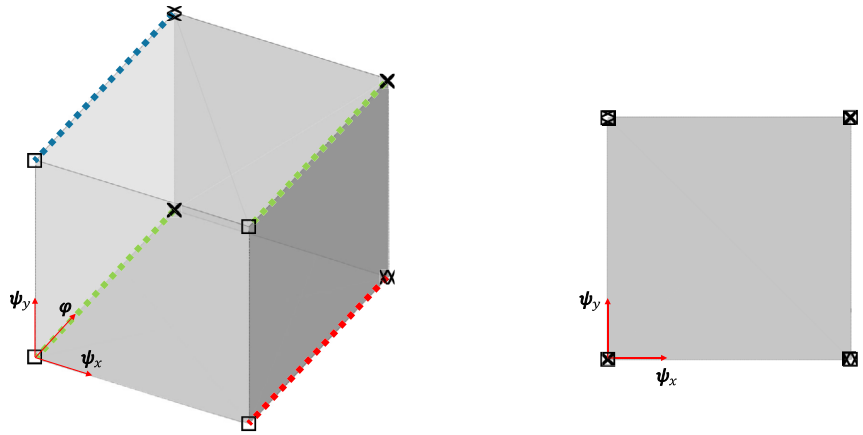


Fig. 10 Cube domain: Studied family placements (top) and material placement for Case I (bottom)

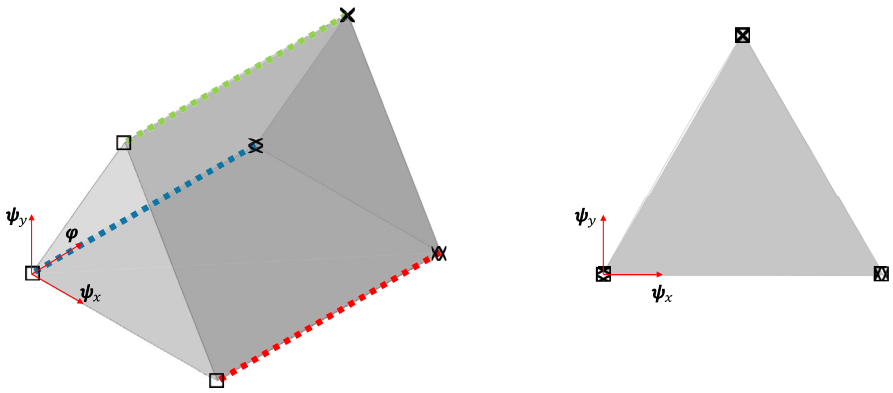


Fig. 11 Triangular prism and material placement

On the other hand, using non-orthogonal domains gives more freedom in setting the number of edges and vertices but requires a more complicated projection technique. The selected domains are illustrated in Fig. 8 and are introduced below.

3.1.1 Square domain

A square domain is the classical orthogonal domain in 2D. We set its size to 1×1 and divide it into 2×9 subdomains in the directions of $\boldsymbol{\psi}$ and $\boldsymbol{\varphi}$, respectively. We then place our materials over the 30 nodes according to their D_{11} values (plotted in Fig. 7), as illustrated in Fig. 9. However, a potential major drawback is the lack of connections between the families, i.e., the resulting interpolation is not necessarily monotone. For instance, it is not possible to go from the red family (at the bottom) to the blue family (at the top) without going through the green family (at the middle), which could result in converging to a bad local minimum.

It is worth mentioning that sorting the families according to their D_{22} values does not affect the optimization process significantly since the curves of D_{11} and D_{22} are reciprocated for Family #2 and Family #3 and same for Family #1. However, sorting the families according to the curves of D_{12} or D_{66} negatively affects the optimization process. This is probably because Family #2 and Family #3 have identical curves for D_{12} and for D_{66}, and when we place them beside each other, we create regions of almost zero-slope and negatively affecting our gradient-based optimization.

On the other hand, sorting the lattices inside each family also affects the optimization progress and the order should be chosen wisely. For instance, sorting the lattices (changing the definition of $\boldsymbol{\varphi}$ in the horizontal axes in Fig. 7) in an ascending or descending order would be the best choice. However, choosing a U-shaped or bell-shaped order gives similar results, but have high sensitivity to initial guess and is not recommended.

3.1.2 Cube domain

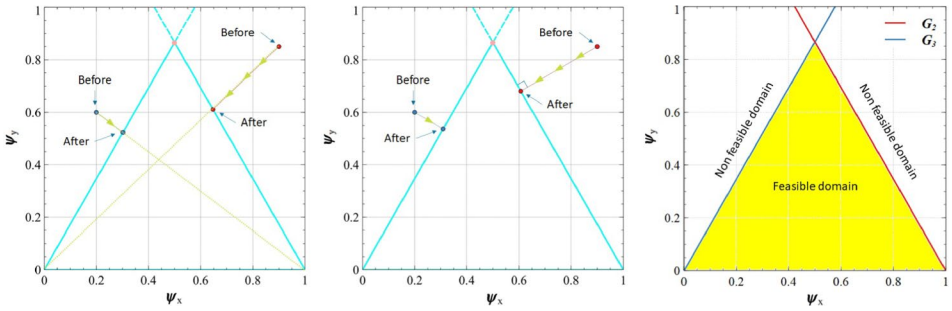
The primary motivation for using a cube domain is to overcome the connectivity issue in the square domain. We take a cube of size $1 \times 1 \times 1$ and place the families over the edges parallel to the $\boldsymbol{\varphi}$-axis as illustrated in Fig. 10(bottom). In the remaining fourth edge, we examine the four possibilities shown in Fig. 10(top). Case I, II, and IV are inspired by the DMO, extended SIMP, and SFP methods (Figs. 1 and 2), respectively, with the black circle representing an artificial family (family #0) that contains only voids.

Since this domain and the following domains are 3D, $\boldsymbol{\psi}$ becomes, from now on, a vector $\boldsymbol{\psi}$ with components in x and y.

3.1.3 Triangular prism

A further improvement compared to the cube domain is achieved using a non-orthogonal triangular prism domain, eliminating the need to duplicate a family or introduce a redundant one. We consider a uniform right triangular prism, cf. Fig. 11. The coordinates of the base triangles are (0, 0, 0), (1, 0, 0), and $(\frac{1}{2}, \frac{\sqrt{3}}{2}, 0)$ and the height of the prism is 1.

During the optimization process, some design points may go outside of the prism, for example, to the coordinate $(1, 1, \boldsymbol{\varphi}_e)$. In order to prevent this, we project the strayed points back to the non-orthogonal domain after each iteration. Since the MMA's box constraints (Eq. (12)) are not enough, from now on, we apply an angular projection, which is illustrated in Fig. 12a. It is worth mentioning that turning off the box constraints did not affect the results. In addition, other more expensive techniques like orthogonal projection or incorporating linear constraints into the optimization process, see Fig. 12b–c, resulted in very similar designs.



(a) Angular Projection (b) Orthogonal Projection (c) Linear constraints

Fig. 12 Techniques for updating design variables within a triangular domain

3.1.4 Triangular pyramid

As seen in Figs. 6 and 7, the three families are identical at $\boldsymbol{\varphi} = 0$ and become more distinct by increasing $\boldsymbol{\varphi}$. This phenomenon makes the replacement of the triangular prism by a triangular pyramid more realistic. Besides, the square-shaped lattice ($\boldsymbol{\varphi} = 0$) does not have to be replicated three times as in the previous domains. Accordingly, we include a triangular pyramid domain with vertices located at $\left(\frac{1}{2}, \frac{\sqrt{3}}{6}, 0\right)$, (0, 0, 1), (1, 0, 1), and $\left(\frac{1}{2}, \frac{\sqrt{3}}{2}, 1\right)$. To avoid singularity issues, the tip node of the pyramid, i.e., point $\left(\frac{1}{2}, \frac{\sqrt{3}}{6}, 0\right)$, is replaced by a very small triangle between points $\left(\frac{1}{2} - \delta, \frac{\sqrt{3}}{6} - \delta, 0\right), \left(\frac{1}{2} + \delta, \frac{\sqrt{3}}{6} - \delta, 0\right), \left(\frac{1}{2}, \frac{\sqrt{3}}{6} + \delta, 0\right)$, with $\delta = 10^{-6}$. Through this modification, the pyramid domain becomes a frustum (prism), which can be mathematically handled as the prism domain before. The domain and the material placement are visualized in Fig. 13.

3.2 Interpolation functions

After deciding how to arrange our materials, the second question is how to interpolate between them. In this section, we compare different interpolation methods like the classical polynomial fitting, ordered SIMP, and smooth single-variable-based interpolation for interpolating $\mathbf{D}_e^{\text{mat}}$ and its derivatives $\partial \mathbf{D}_e^{\text{mat}} / \partial \boldsymbol{\psi}_e$ and $\partial \mathbf{D}_e^{\text{mat}} / \partial \boldsymbol{\varphi}_e$ in Eqs. (8), (9) and (10). Figure 14 gives a general overview of the methods, further presented in the following subsections.

We start with introducing the methods used to interpolate the stiffness of isotropic materials with respect to pseudo density, i.e., $E_e(\boldsymbol{\rho}_e)$ in 1D domains. Here, we interpolate each scalar component of the stiffness matrices of our orthotropic materials with respect to geometry design variables, i.e., $\mathbf{D}_e^{\text{mat}}(\boldsymbol{\psi}_e, \boldsymbol{\varphi}_e)$ and also extend the methods to 2D and 3D domains.

3.2.1 Polynomial fitting

This classical method is featured by its continuous function and derivatives; see Fig. 14a. It approximates the modulus as

$$E_e^{\text{poly}}(\boldsymbol{\rho}_e) = \sum_{i=0}^{n} a_i {\boldsymbol{\rho}_e}^i, \tag{13}$$

where a_i are the fitting coefficients; and n is the fitting degree. Here, we have two directions to fit. In the direction of $\boldsymbol{\psi}$, we choose n such that all families are interpolated, i.e., the coefficient of determination R^2 equals one, which means $n_{\boldsymbol{\psi}} = 2$ for the square domain and $n_{\boldsymbol{\psi}} = 1$ for the other ones.

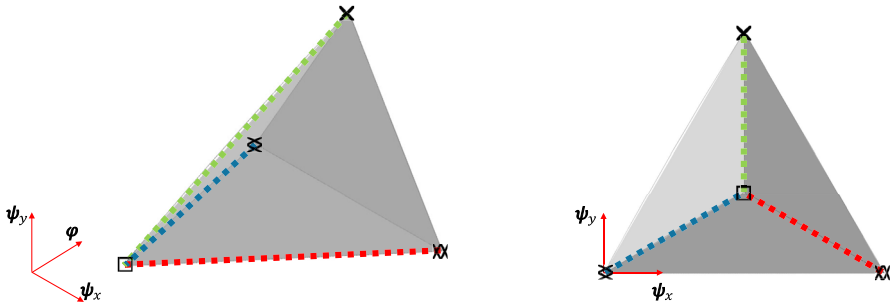
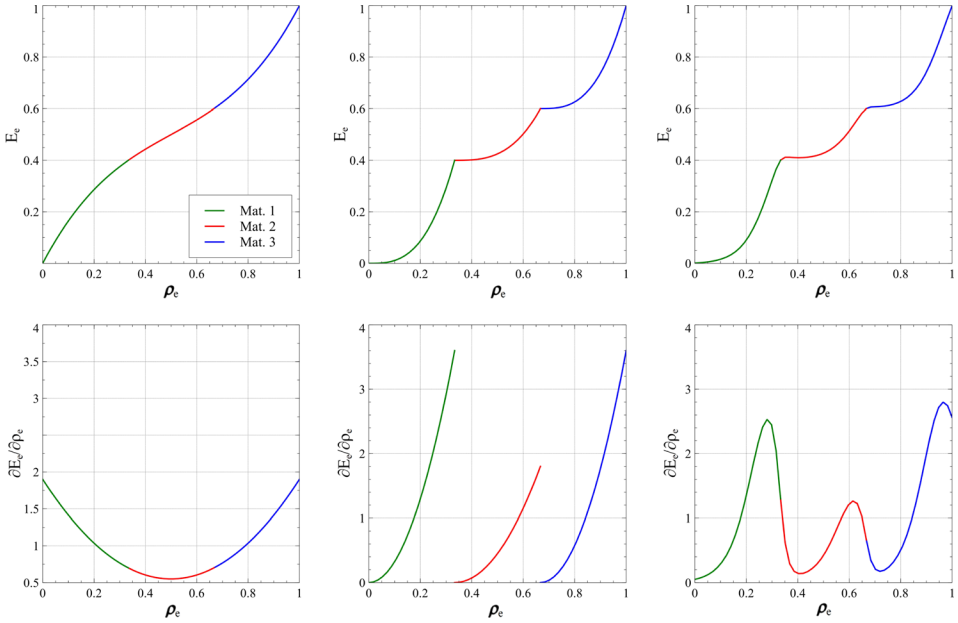


Fig. 13 Triangular pyramid



(a) Polynomial fitting (b) Modf. ordr. SIMP (c) Smth. sngl. var.

Fig. 14 Interpolation functions (top) and their first derivatives (bottom) with $n = p = \beta = 3$ and $(\rho, E) \in \{(0,0), (\frac{1}{3}, \frac{2}{5}), (\frac{2}{3}, \frac{3}{5}), (1,1)\}$

In the direction of $\boldsymbol{\varphi}$, we set $n_{\boldsymbol{\varphi}} = 3$, which results in good fittings with all $R^2 \geq 0.999$.

This method is not favored in MMTO due to its lack of penalization, which means the optimized design will have very scattered design points (non-physical material points) rather than converging to the predefined materials. Accordingly, we project $\boldsymbol{\psi}$ and $\boldsymbol{\varphi}$ to their nearest node (material) after the convergence.

3.2.2 Modified ordered SIMP

This method, cf. Fig. 14b, was developed by Zuo and Saitou (2016) and was improved by Augusto and Palma (2022) as an extension of the modified SIMP method to account for multiple materials. It interpolates the modulus between material m and material $m+1$ with $\boldsymbol{\rho}_{m+1} > \boldsymbol{\rho}_m$ as:

$$E_e^{\text{ordr}}(\boldsymbol{\rho}_e) = \begin{cases} a\left(\frac{\rho_e - c}{\rho_{m+1} - \rho_m}\right)^p + d, & E_{m+1} \geq E_m \\ a\left(\frac{c - \rho_e}{\rho_{m+1} - \rho_m}\right)^p + d, & E_{m+1} < E_m, \end{cases} \tag{14}$$

where p is a penalty factor (we choose $p = 3$), and

$$a = \begin{cases} E_{m+1} - E_m, & E_{m+1} \geq E_m \\ E_m - E_{m+1}, & E_{m+1} < E_m \end{cases}$$

$$c = \begin{cases} \boldsymbol{\rho}_m, & E_{m+1} \geq E_m \\ \boldsymbol{\rho}_{m+1}, & E_{m+1} < E_m \end{cases}$$

$$d = \begin{cases} E_m, & E_{m+1} \geq E_m \\ E_{m+1}, & E_{m+1} < E_m. \end{cases}$$

3.2.3 Smooth single-variable-based interpolation

To make the first derivatives of the ordered SIMP method continuous, Dinh et al. (2024) proposed the two following modifications:

1. Replacing the penalized term $\left(\frac{\rho_e - c}{\rho_{m+1} - \rho_m}\right)^p$, in Eq. (14), with a continuous Heaviside function $H(\boldsymbol{\rho}_e)$ as

$$E_e^{\text{heav}}(\boldsymbol{\rho}_e) = (E_{m+1} - E_m) H_e(\boldsymbol{\rho}_e) + E_m, \tag{15}$$

with

$$H_e(\boldsymbol{\rho}_e) = \frac{\tanh \beta\tilde{\eta} + \tanh \beta\left(\frac{\rho_e - \rho_m}{\rho_{m+1} - \rho_m} - \tilde{\eta}\right)}{\tanh \beta\tilde{\eta} + \tanh \beta(1 - \tilde{\eta})},$$

$$\tilde{\eta} = \begin{cases} 1 - \eta, & E_{m+1} \geq E_m \\ \eta, & E_{m+1} < E_m, \end{cases}$$

where $\beta > 0$ is a penalization parameter and $\eta \in (0, 0.5)$ is a design parameter.

2. Smoothing the interpolated modulus, in Eq. (15), using the global approximations of Wu et al. (2015):

$$E_e^{\text{smth}}(\boldsymbol{\rho}_e) = \sum_{i=2}^{M} F_i(\boldsymbol{\rho}_e) - \sum_{i=2}^{M-1} E_i^{\text{heav}}(\boldsymbol{\rho}_e), \tag{16}$$

with:

$$F_i(\boldsymbol{\rho}_e) = \frac{1}{2}(E_i^{\text{heav}} + E_{i-1}^{\text{heav}}) + \left(\frac{(\boldsymbol{\rho}_e - \boldsymbol{\rho}_i)}{2\sqrt{(\boldsymbol{\rho}_e - \boldsymbol{\rho}_i)^2 + \epsilon}}\right)(E_i^{\text{heav}} - E_{i-1}^{\text{heav}}),$$

where $\epsilon > 0$ is a smoothing parameter.
Here, we select $\beta = 3$, $\eta = 0.1$, and $\epsilon = 0.001$.

3.2.4 Extensions to 2D and 3D using Coons patch

In the previous subsections, we presented the interpolation methods used in this study: polynomial fitting, modified ordered SIMP and smooth single-variable-based interpolation. As we consider the 2D and 3D domains presented in Sect. 3.1, our interpolation functions need to be extended to 2D and 3D. For this, we use the concept of Coons patches (Coons 1967), which constructs a surface or volumetric patch by bilinearly or trilinearly blending the given boundary of the surface or of the volumetric patch, respectively. Using the concept of Coons patches is a key feature of our paper, as we keep the number of design variables depending only on the dimensionality of the interpolation domain. In addition, Coons patch is a convenient approach for extending univariate interpolation (defined on the boundary of the interpolation domain) to multivariate interpolation.

Coons patches have already been used in isogeometric shape optimization to describe the geometry (Qian and Sigmund 2011). Here, we use Coons patches to build the interpolation of material properties, and we extend it to higher dimensional domains.

As an example, we give here an illustration for interpolating over a square domain $\mathcal{D}^{\text{sqr}}$ having the micro-structures with these elasticity tensors $\mathbf{D}_{\text{BL}}$, $\mathbf{D}_{\text{BR}}$, $\mathbf{D}_{\text{TR}}$, and $\mathbf{D}_{\text{TL}}$ over its bottom-left, bottom-right, top-right, and top-left vertices, respectively.

For finding the interpolated elasticity tensor $\mathbf{D}^{\text{coons}}$ corresponding to the design point $(\boldsymbol{\psi}, \boldsymbol{\varphi}) \in \mathcal{D}^{\text{sqr}}$, we first interpolate over the four edges using the designated univariate function to get $\mathbf{D}_{\text{L}}(\boldsymbol{\varphi})$, $\mathbf{D}_{\text{R}}(\boldsymbol{\varphi})$, $\mathbf{D}_{\text{T}}(\boldsymbol{\psi})$, and $\mathbf{D}_{\text{B}}(\boldsymbol{\psi})$, and then find the interpolated $\mathbf{D}^{\text{coons}}$ applying the Coons patches concept, as

$$\mathbf{D}^{\text{coons}}(\boldsymbol{\psi}, \boldsymbol{\varphi}) = \mathbf{D}_{\text{LR}} + \mathbf{D}_{\text{BT}} - \mathbf{D}_{\text{O}},$$

with

$$\begin{aligned}
\mathbf{D}_{\text{LR}} &= (1 - \boldsymbol{\psi})\,\mathbf{D}_{\text{L}}(\boldsymbol{\varphi}) + \boldsymbol{\psi}\,\mathbf{D}_{\text{R}}(\boldsymbol{\varphi}),\\
\mathbf{D}_{\text{BT}} &= (1 - \boldsymbol{\varphi})\,\mathbf{D}_{\text{B}}(\boldsymbol{\psi}) + \boldsymbol{\varphi}\,\mathbf{D}_{\text{T}}(\boldsymbol{\psi}),\\
\mathbf{D}_{\text{O}} &= (1 - \boldsymbol{\psi})\big[(1 - \boldsymbol{\varphi})\,\mathbf{D}_{\text{BL}} + \boldsymbol{\varphi}\,\mathbf{D}_{\text{TL}}\big]\\
&\quad + \boldsymbol{\psi}\big[(1 - \boldsymbol{\varphi})\,\mathbf{D}_{\text{BR}} + \boldsymbol{\varphi}\,\mathbf{D}_{\text{TR}}\big].
\end{aligned}$$

The other explicit formulas that were implemented besides the Python codes are given in Appendices A and B, respectively.

4 Examples and discussion

In this section, we present our optimization problem and discuss the results obtained using different interpolation domains and functions.

4.1 Design of a Messerschmitt-Bölkow-Blohm beam

A schematic for the MBB beam problem is shown in Fig. 15. We consider symmetry and use dimensionless quantities. The beam has a half-span and a height of 0.50 and 0.25, respectively, and is modeled by using 50×25 equally sized plane-strain square elements.

To comply with the FE fundamental rules, the point load is replaced with a distributed unit pressure over a strip of 0.10. In addition, the penalty approach is applied to enforce the average displacement over a span of 0.01 to be zero, rather than having a point support at the lower right corner (Sigmund 2022).

4.2 Results

The volume fraction is set to $V_{\text{f}} = 40\%$, and the MBB problem is solved by using all possible material models [i.e., combinations of interpolation domains (Sect. 3.1) and interpolation functions (Sect. 3.2)]. Generally, all models show typical behaviors where compliance drops dramatically during the first 25 iterations and gradually reduces until convergence. Similarly, the volumes fluctuate at the beginning, where many gray elements exist, but then stabilize until the iteration processes converge. For brevity, we only present the iteration histories using the triangular prism; see Fig. 16.

Figures 17, 18, and 19 show the optimized designs for all interpolation domains by using polynomial fitting, modified ordered SIMP method, and smooth single-variable

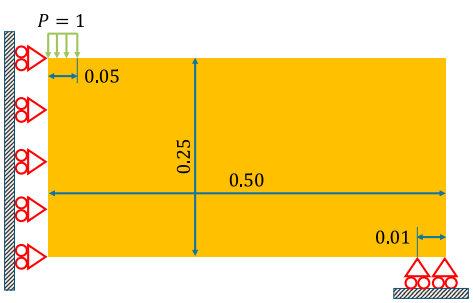


Fig. 15 Elasticity problem: MBB beam

interpolation, respectively. The optimized values of $\boldsymbol{\rho}$, $\boldsymbol{\psi}$, and $\boldsymbol{\varphi}$ are plotted in the upper, middle, and lower subplots, respectively.

Except in Figs. 18f–g and 19f–g, the designs have uniform distributions of lattices which also coincide with the directions of the principal stresses. For instance, the lower region of the design domain, where the shear stress is low, is filled with beige $\boldsymbol{\varphi}$ which corresponds to the square-shaped lattice (see Fig. 6) regardless of the color of $\boldsymbol{\psi}$. On the contrary, the shear stress is high in the middle of the design domain, making the ×-shaped lattice (red $\boldsymbol{\psi}$ and dark brown $\boldsymbol{\varphi}$) more favored. At the left and right regions, where the vertical load and support exist, some material models have converged to the ⋇-shaped lattice (red $\boldsymbol{\psi}$ and dark brown $\boldsymbol{\varphi}$) due to its dominant D_{22}^{mat} value, where the ⨯-shaped lattice (blue $\boldsymbol{\psi}$ and brown $\boldsymbol{\varphi}$) is occupying the upper region of the design domain where the normal stress σ_{xx} (horizontal) is high.

Taking the optimized design of Fig. 17a as an example, we plot the corresponding two-scale structure in Fig. 20. As we see, the distribution of the lattices is uniform and the connectivity between the unit cells is generally good

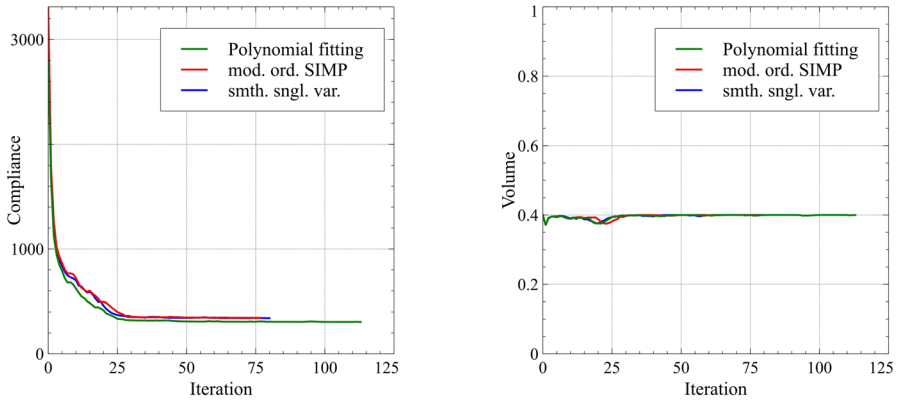
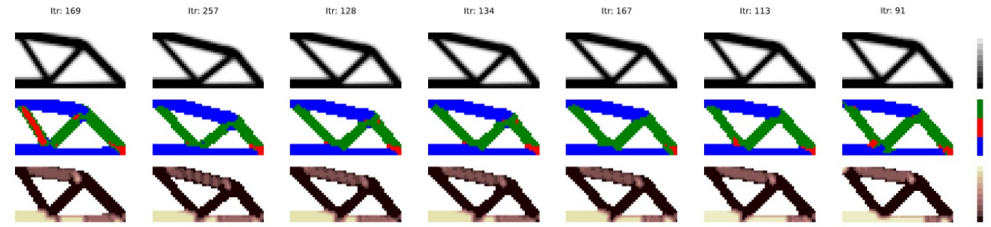
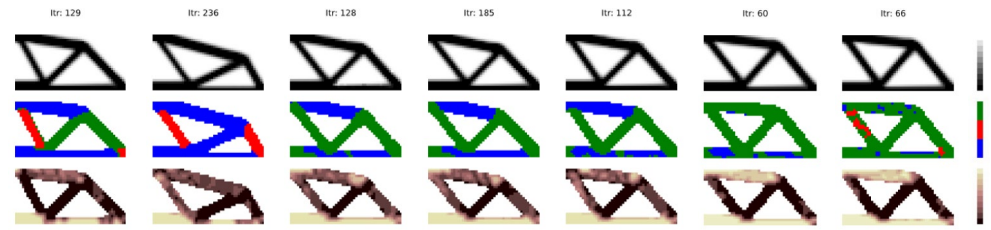


Fig. 16 Iteration history when using the triangular prism domain



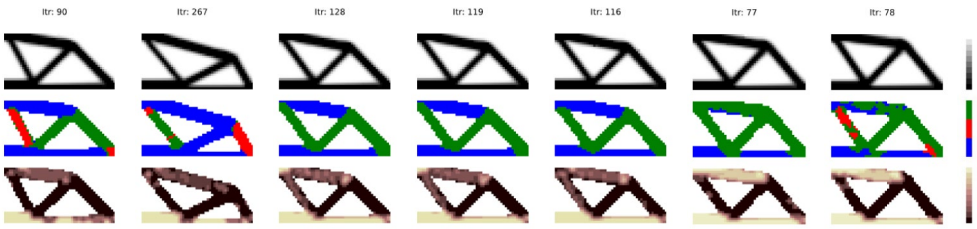
(a) square C = 310.94 (b) cube I C = 312.17 (c) cube II C = 310.99 (d) cube III C = 310.43 (e) cube IV C = 308.13 (f) prism C = 310.58 (g) pyramid C = 317.40 .

Fig. 17 Optimized designs using polynomial fitting method and different interpolation domains. The optimized compliance values are given below the names of the interpolation domains



(a) square C = 313.71 (b) cube I C = 326.96 (c) cube II C = 324.00 (d) cube III C = 321.32 (e) cube IV C = 320.34 (f) prism C = 343.01 (g) pyramid C = 342.55 .

Fig. 18 Optimized designs using Ordered SIMP method and different interpolation domains. The optimized compliance values are given below the names of the interpolation domains



(a) square C = 309.58 (b) cube I C = 319.35 (c) cube II C = 317.23 (d) cube III C = 316.93 (e) cube IV C = 313.46 (f) prism C = 340.57 (g) pyramid C = 338.04 .

Fig. 19 Optimized designs using the smooth single-variable interpolation method and different interpolation domains. The optimized compliance values are given below the names of the interpolation domains

because our lattices have good connectives at the boundaries except the -shaped, and ⨳-shaped lattices as they match only at the corners. However, when it comes to manufacturing the optimized structure, the small regions with doubtful conductivity might be slightly changed to their neighbor lattices, i.e., changing $\boldsymbol{\varphi}_e$ from 1 (lattice without outer frame) to 0.9 (lattice with a thin outer frame). This of course results in a small negligible compromise in compliance, as our demonstrated example shows almost zero reduction. Another possible solution is to include kinematical connectors as done in Hu et al. (2021) and Li et al. (2017).

Back to the lattices heterogeneity seen in Figs. 18f–g and 19f–g, the reason could be raised to the penalization ($p = \beta = 3$) used in the modified ordered SIMP and smooth single-variable interpolations. In the following subsection, we study the effect of penalizing $\boldsymbol{\psi}$ and $\boldsymbol{\varphi}$ in detail.

4.3 Effect of the penalization

In this subsection, we study the effect of the penalization factors p and β. The main goal for introducing them was to make the non-physical intermediate materials (gray) between any two physical materials (black and white) less efficient, i.e., having less stiffness-to-weight ratios.

In this study, we applied the penalization to $\boldsymbol{\psi}$ and $\boldsymbol{\varphi}$ rather than $\boldsymbol{\rho}$, as in the original papers (Zuo and Saitou 2016; Augusto and Palma 2022; Dinh et al. 2024). The optimized compliance of the four cases resulting from penalizing and not penalizing $\boldsymbol{\psi}$ and $\boldsymbol{\varphi}$ are compared in Fig. 21. On the one hand, it is found that penalizing $\boldsymbol{\psi}$ leads to bad minima due to the fact that it represents no physical property. On the other hand, penalizing $\boldsymbol{\varphi}$, which

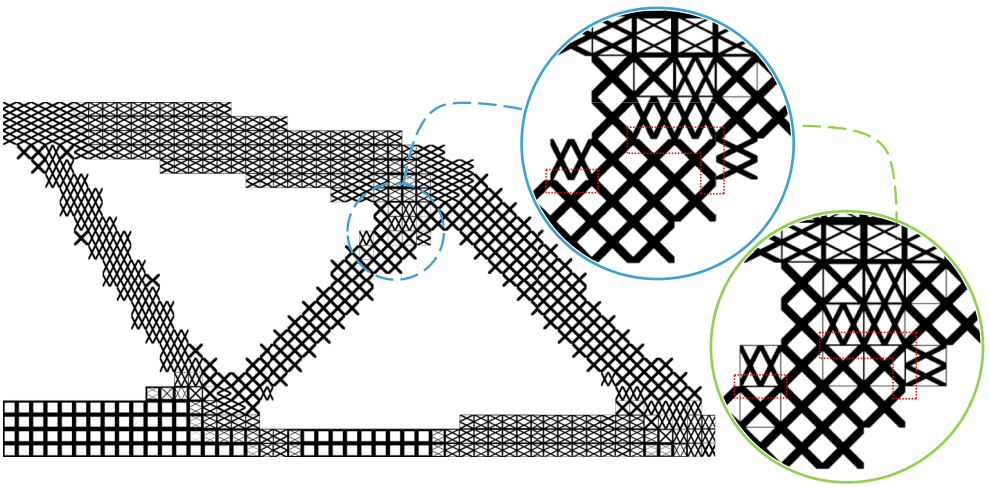


Fig. 20 The corresponding microstructure for the optimized design seen in Fig. 17a. The blue zooming bubble shows some regions with doubtful connectivity surrounded by red dotted lines which has been manually modified (by changing $\boldsymbol{\varphi}_e$ from 1 to 0.9) as shown in the green zooming bubble. Compliance before and after the modification are 310.94 and 310.43, respectively. Further improved connectivity can be achieved by changing $\boldsymbol{\varphi}_e$ from 1 to 0.8, instead

is somehow indicates shear stiffness (D_{12}^{mat} and D_{66}^{mat}), results in bad minima too. We attribute this phenomenon to the fact that $\boldsymbol{\psi}$ and $\boldsymbol{\varphi}$ are coupled, i.e., $\mathbf{D}_e^{\text{mat}}(\boldsymbol{\psi}_e, \boldsymbol{\varphi}_e)$ is not separable, and conclude that both of $\boldsymbol{\psi}$ and $\boldsymbol{\varphi}$ should not be penalized. For comparability, we plot the designs obtained without penalization in Figs. 22 and 23.

4.4 Summary

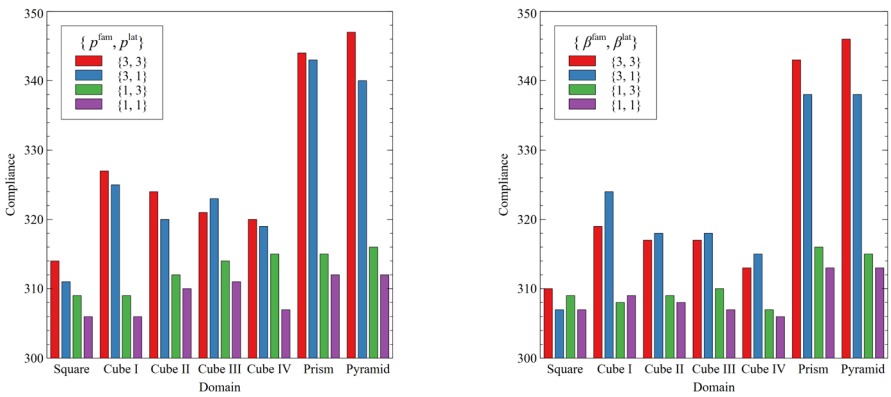
To sum up, we compared three different interpolation methods with distinct features, as summarized in Table 1, in optimizing an FGL structure illustrated in an MBB beam problem. Our study found that the penalizations offered by both ordered SIMP and smooth single-variable-based interpolations do not necessarily improve the optimization process when all materials have similar weights. Besides, the derivatives' discontinuity found in the ordered SIMP method did not cause any numerical instabilities during the optimization process because hitting exactly the material points (during the design variable optimization) is not a common event. Moreover, the local smoothening offered by the smooth single-variable-based interpolation was computationally expensive due to the fact that Eq. (16) depends on the number of lattices, which is ten in our case.

On the other hand, 3D interpolation domains offered more design freedom for switching between the families by increasing only one design variable per mesh element. However, this resulted in the same behavior of sorting the families linearly in a proper way considering their mechanical properties. This was true also using non-orthogonal domains. Table 2 summarizes all results obtained using the studied interpolation domains and functions.

5 Conclusion

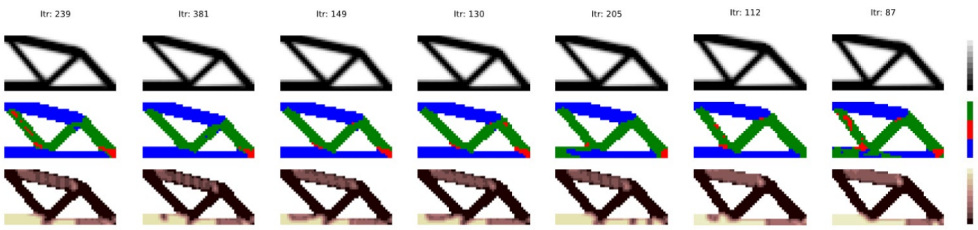
In this paper, we compared the optimization behavior of optimizing FGL composed from 28 lattices having similar densities by using MMTO models built from several combinations of material interpolation domains and interpolation functions. The studied interpolation domains consist of 2D and 3D domains in addition to orthogonal and non-orthogonal ones. For the interpolation functions, the collection consists of methods having smooth, non-smooth, and smoothened functions, as well as continuous and non-continuous derivatives.

We generated different material models by having the square, cube, triangular prism, and triangular pyramid



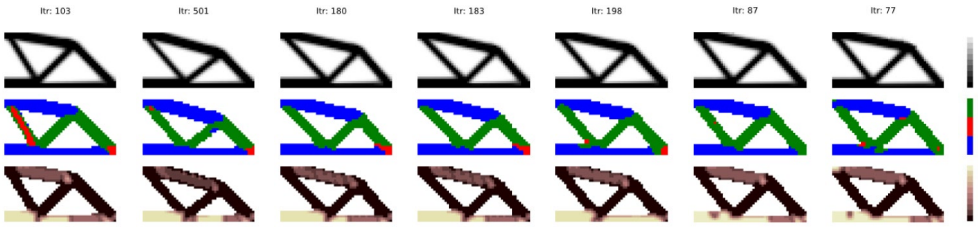
modf. ordr. SIMP smth. sngl. var.

Fig. 21 optimized compliance when using different penalization factors (p & β)



(a) square C = 306.12 (b) cube I C = 305.61 (c) cube II C = 309.79 (d) cube III C = 310.58 (e) cube IV C = 306.67 (f) prism C = 314.08 (g) pyramid C = 315.67 .

Fig. 22 Optimized designs using Ordered SIMP (without penalization) and different domains. The optimized compliance values are given below the names of the interpolation domains



(a) square C = 307.02 (b) cube I C = 308.68 (c) cube II C = 307.94 (d) cube III C = 307.48 (e) cube IV C = 305.73 (f) prism C = 312.32 (g) pyramid C = 318.84 .

Fig. 23 Optimized designs using the smooth single-variable (without penalization) and different domains. The optimized compliance values are given below the names of the interpolation domains

Table 1 Qualitative comparison between the interpolation functions

Feature	Interpolation method		
	Polynomial fitting	Modf. ordr. SIMP	Smth. sngl. var
Penalization	×	√	√
Continuous function	√	√	√
Smooth function	√	×	√
Continuous derivative	√	×	√
Smooth derivative	√	×	√

domains on the one hand and polynomial fitting, modified ordered SIMP, and smooth single-variable-based interpolation functions on the other hand. As a novelty, we used the concept of Coons patches to extend the original 1D interpolation functions to 2D and 3D to match our interpolation domains.

In all tested models, the number of design variables does not depend on the number of candidate materials but on the dimensionality of the interpolation domain, i.e., 2D or 3D. We compared their optimization behaviors by solving an MBB beam problem. The results showed that using customized and more complicated domains like triangular prism and triangular pyramid does not necessarily lead to better designs. Although they offer direct connections between the families by adding only one design variable as well as the square domain, the gain was not significant. The same conclusion applies to the interpolation functions where the smooth single-variable interpolation requires an additional computation cost (depending on the number of used materials) for smoothening the kinks that arise due to the multi-fitting.

In our case, the wise arrangement of categorizing our 28 micro-structures in three families according to their geometrical features, sorting the micro-structures of each family in an ascending/descending order, and arranging the three families according to the first component of their elasticity tensors, enabled us to utilize the use of the low cost combination of a 2D square domain together with piecewise-defined function to get similar to those of other combinations. However, this might not be the case anymore when working with micro-structures of variable densities, or when working with more complex formulations such as designing compliance mechanisms, which will be addressed in future work.

Table 2 Optimized compliance for the studied interpolation domains and functions

		Interpolation function				
		Polynomial fitting	Modf. ordr. SIMP		Smth. sngl. var.	
			$p = 3$	$p = 1$	$\beta = 3$	$\beta = 1$
Domain	Square	310.94	313.71	306.12	309.58	307.02
	Cube—I	312.17	326.96	305.61	319.35	308.68
	Cube—II	310.99	324.00	309.79	317.23	307.94
	Cube—III	310.43	321.32	310.58	316.93	307.48
	Cube—IV	308.13	320.34	306.67	313.46	305.73
	Tri. Prism	310.58	343.01	314.08	340.57	312.32
	Tri. Pyramid	317.40	342.55	315.67	338.04	318.84

Appendix A: Coons patches: explicit formulas

In this appendix, we give in details the extensions of the interpolation functions (Sect. 3.2) from the 1D line domain to the 2D square domain and 3D cube, triangular prism, and triangular pyramid domains using the concept of Coons patches. The following subsections show the equations used for interpolating $\mathbf{D}_e^{\text{mat}}$ inside each domain. For the sake of readability, we give the equations in (x, y, z) coordinates instead of $(\boldsymbol{\psi}^x, \boldsymbol{\psi}^y, \boldsymbol{\varphi})$.

Appendix A.1: Square domain

The interpolation over a square domain (Fig. 24a) is done via

$$\mathbf{D}^{\text{coons}}(x,y) = \mathbf{S}_{xy}^{\text{sqr}} = \mathbf{S}_x^{\text{sqr}} + \mathbf{S}_y^{\text{sqr}} - \mathbf{S}^{\text{o}}, \tag{17}$$

with

$$\begin{aligned}
\mathbf{S}_x^{\text{sqr}} &= (1-x)\mathbf{C}_{0y}(y) + x\mathbf{C}_{1y}(y),\\
\mathbf{S}_y^{\text{sqr}} &= (1-y)\mathbf{C}_{x0}(x) + y\mathbf{C}_{x1}(x),\\
\mathbf{S}^{\text{o}} &= (1-x)\big[(1-y)\mathbf{P}_{00} + y\mathbf{P}_{01}\big]\\
&\quad + x\big[(1-y)\mathbf{P}_{10} + y\mathbf{P}_{11}\big],
\end{aligned}$$

where the points $\mathbf{P}$ are our materials (i.e., $\mathbf{D}_e^{\text{mat}}(\boldsymbol{\psi}_e, \boldsymbol{\varphi}_e)$); and the curves $\mathbf{C}$ are the interpolation functions presented in Sect. 3.2. $\mathbf{P}$ and $\mathbf{C}$ are illustrated in Fig. 24a below, for all interpolation domains.

Appendix A.2: Triangle domain

In this study, we do not use a triangle domain, but we present it for completeness and to facilitate the readability of the prism and pyramid domains subsection.

The property over a triangle domain (Fig. 24b) is interpolated as Barnhill et al. (1973)

$$\mathbf{D}^{\text{coons}}(u,v) = \mathbf{S}_{\text{uv}}^{\text{tri}} = \frac{1}{2}\mathbf{S}_{\text{u}} + \frac{1}{2}\mathbf{S}_{\text{v}} + \frac{1}{2}\mathbf{S}_{\text{w}} - \frac{1}{2}\mathbf{S}_{\text{o}}, \tag{18}$$

with

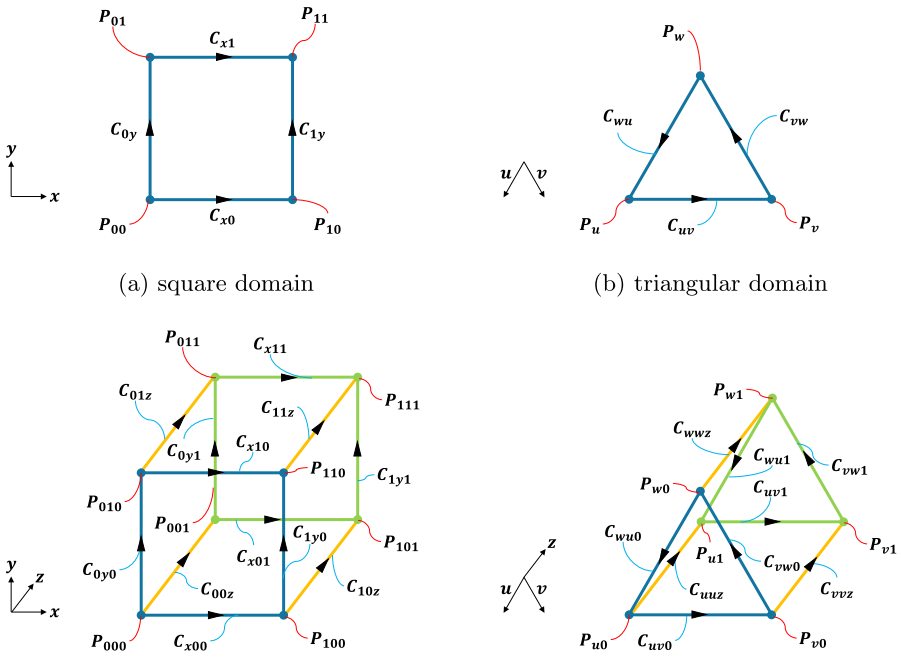
$$\begin{aligned}
\mathbf{S}_{\text{u}} &= (1-\widetilde{u})\mathbf{C}_{\text{vw}}(1-v) + \widetilde{u}\mathbf{C}_{\text{uv}}(v),\\
\mathbf{S}_{\text{v}} &= \widetilde{v}\mathbf{C}_{\text{uv}}(1-u) + (1-\widetilde{v})\mathbf{C}_{\text{wu}}(u),\\
\mathbf{S}_{\text{w}} &= (1-\widetilde{w})\mathbf{C}_{\text{wu}}(1-w) + \widetilde{w}\mathbf{C}_{\text{vw}}(w),
\end{aligned}$$

and

$$\mathbf{S}_{\text{o}} = u\mathbf{P}_{\text{u}} + v\mathbf{P}_{\text{v}} + w\mathbf{P}_{\text{w}},$$

where u, v, and w are the Barycentric coordinates; and $\widetilde{u}$, $\widetilde{v}$, and $\widetilde{w}$ are calculated as

$$\begin{aligned}
\widetilde{u} &= \frac{u}{u+w},\\
\widetilde{v} &= \frac{v}{v+w},\\
\widetilde{w} &= \frac{v}{u+v}.
\end{aligned}$$



(a) square domain

(b) triangular domain

(c) cube domain

(d) triangular prism

Fig. 24 Points and curves used to evaluate the Coons patches

Appendix A.3: Cube domain

Similarly to the square domain, the interpolation over a cube domain (Fig. 24c) reads

$$\mathbf{D}^{\text{coons}}(x,y,z) = \mathbf{V}^{\text{cube}}_{x} + \mathbf{V}^{\text{cube}}_{y} + \mathbf{V}^{\text{cube}}_{z} - \mathbf{V}^{\text{cube}}_{yz} - \mathbf{V}^{\text{cube}}_{zx} - \mathbf{V}^{\text{cube}}_{xy} + \mathbf{V}^{\text{o}}, \tag{19}$$

with

$$\begin{aligned}
\mathbf{V}_{x} &= (1-x)\mathbf{S}^{\text{sqr}}_{0yz} + x\mathbf{S}^{\text{sqr}}_{1yz},\\
\mathbf{V}_{y} &= (1-y)\mathbf{S}^{\text{sqr}}_{x0z} + y\mathbf{S}^{\text{sqr}}_{x1z},\\
\mathbf{V}_{z} &= (1-z)\mathbf{S}^{\text{sqr}}_{xy0} + z\mathbf{S}^{\text{sqr}}_{xy1},
\end{aligned}$$

where $\mathbf{S}^{\text{sqr}}_{0yz}, \mathbf{S}^{\text{sqr}}_{1yz}, \mathbf{S}^{\text{sqr}}_{x0z}, \mathbf{S}^{\text{sqr}}_{x1z}$, and $\mathbf{S}^{\text{sqr}}_{xy0}, \mathbf{S}^{\text{sqr}}_{xy1}$ are the square surfaces $\mathbf{S}^{\text{sqr}}_{yz}, \mathbf{S}^{\text{sqr}}_{xz}$, and $\mathbf{S}^{\text{sqr}}_{xy}$, at the faces $x = 0$ and $x = 1$, $y = 0$ and $y = 1$, and $z = 0$ and $z = 1$, respectively, calculated from Eq. (17), and

$$\begin{aligned}
\mathbf{V}_{yz} &= (1-y)\big[(1-z)\mathbf{C}_{x00} + z\mathbf{C}_{x01}\big] + y\big[(1-z)\mathbf{C}_{x10} + z\mathbf{C}_{x11}\big],\\
\mathbf{V}_{zx} &= (1-z)\big[(1-x)\mathbf{C}_{0y0} + x\mathbf{C}_{1y0}\big] + z\big[(1-x)\mathbf{C}_{0y1} + x\mathbf{C}_{1y1}\big],\\
\mathbf{V}_{xy} &= (1-x)\big[(1-y)\mathbf{C}_{00z} + y\mathbf{C}_{01z}\big] + x\big[(1-y)\mathbf{C}_{10z} + y\mathbf{C}_{11z}\big],
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{V}^{\text{o}} = (1-x)\big\{&(1-y)\big[(1-z)\mathbf{P}_{000} + z\mathbf{P}_{001}\big]\\
&+ y\big[(1-z)\mathbf{P}_{010} + z\mathbf{P}_{011}\big]\big\}\\
+ x\big\{&(1-y)\big[(1-z)\mathbf{P}_{100} + z\mathbf{P}_{101}\big]\\
&+ y\big[(1-z)\mathbf{P}_{110} + z\mathbf{P}_{111}\big]\big\}.
\end{aligned}$$

Appendix A.4: Prism and pyramid domains

For both prism and pyramid domains (Fig. 24d), the interpolation is done through

$$\mathbf{D}^{\text{coons}}(x,y,z) = \mathbf{V}^{\text{prsm}}_{\text{u}} + \mathbf{V}^{\text{prsm}}_{\text{v}} + \mathbf{V}^{\text{prsm}}_{\text{w}} + V^{\text{prsm}}_{z} - \frac{1}{2}\mathbf{V}^{z} \tag{20}$$

$$- \mathbf{V}^{\text{prsm}}_{\text{uz}} - \mathbf{V}^{\text{prsm}}_{\text{vz}} - \mathbf{V}^{\text{prsm}}_{\text{wz}} + \frac{1}{2}\mathbf{V}^{\text{o}}, \tag{21}$$

with

$$\begin{aligned}
\mathbf{V}^{\text{prsm}}_{\text{u}} &= \frac{1}{2}\big[(1-\widetilde{u})\mathbf{S}^{\text{sqr}}_{\text{vw}}(1-v,z) + \widetilde{u}\mathbf{S}^{\text{sqr}}_{\text{uv}}(v,z)\big],\\
\mathbf{V}^{\text{prsm}}_{\text{v}} &= \frac{1}{2}\big[(1-\widetilde{v})\mathbf{S}^{\text{sqr}}_{\text{wu}}(1-w,z) + \widetilde{v}\mathbf{S}^{\text{sqr}}_{\text{vw}}(w,z)\big],\\
\mathbf{V}^{\text{prsm}}_{\text{w}} &= \frac{1}{2}\big[(1-\widetilde{w})\mathbf{S}^{\text{sqr}}_{\text{uv}}(u,z) + \widetilde{w}\mathbf{S}^{\text{sqr}}_{\text{wu}}(1-u,z)\big],\\
\mathbf{V}^{\text{prsm}}_{z} &= (1-z)\mathbf{S}^{\text{tri}}_{xy0}(u,v) + z\mathbf{S}^{\text{tri}}_{xy1}(u,v),
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{V}^{\text{prsm}}_{\text{uz}} &= \frac{1}{2}(1-\widetilde{u})\big[(1-z)\mathbf{C}_{\text{vw0}} + z\mathbf{C}_{\text{vw1}}\big] + \frac{1}{2}\widetilde{u}\big[(1-z)\mathbf{C}_{\text{uv0}} + z\mathbf{C}_{\text{uv1}}\big],\\
\mathbf{V}^{\text{prsm}}_{\text{vz}} &= \frac{1}{2}(1-\widetilde{v})\big[(1-z)\mathbf{C}_{\text{wu0}} + z\mathbf{C}_{\text{wu1}}\big] + \frac{1}{2}\widetilde{v}\big[(1-z)\mathbf{C}_{\text{vw0}} + z\mathbf{C}_{\text{vw1}}\big],\\
\mathbf{V}^{\text{prsm}}_{\text{wz}} &= \frac{1}{2}(1-\widetilde{w})\big[(1-z)\mathbf{C}_{\text{uv0}} + z\mathbf{C}_{\text{uv1}}\big] + \frac{1}{2}\widetilde{w}\big[(1-z)\mathbf{C}_{\text{wu0}} + z\mathbf{C}_{\text{wu1}}\big],
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{V}^{z} &= u\mathbf{C}_{\text{uuz}} + v\mathbf{C}_{\text{vvz}} + w\mathbf{C}_{\text{wwz}},\\
\mathbf{V}^{\text{o}} &= (1-z)\big(u\mathbf{P}_{\text{u0}} + v\mathbf{P}_{\text{v0}} + w\mathbf{P}_{\text{w0}}\big) + z\big(u\mathbf{P}_{\text{u1}} + v\mathbf{P}_{\text{v1}} + w\mathbf{P}_{\text{w1}}\big),
\end{aligned}$$

where the square surfaces $\mathbf{S}^{\text{sqr}}_{\text{wu}}$, $\mathbf{S}^{\text{sqr}}_{\text{uv}}$, and $\mathbf{S}^{\text{sqr}}_{\text{vw}}$ are evaluated using Eq. (17) at the points $(\widetilde{u}, z)$, $(\widetilde{v}, z)$, and $(\widetilde{w}, z)$, respectively; and $\mathbf{S}^{\text{tri}}_{xy0}$ and $\mathbf{S}^{\text{tri}}_{xy1}$ are the triangular surfaces $\mathbf{S}^{\text{tri}}_{xy}$ at the faces $z = 0$ and $z = 1$, respectively, evaluated by Eq. (18).

Appendix B: Coons patches: python codes

Python codes used in this paper for extending univariate functions to square, equilateral triangle, cube, and triangular prism and triangular pyramid domains by concepts of Coons patches are made available on Ali (2025) for public use.

Acknowledgements The authors would like to thank the anonymous reviewers for their comments that helped to improve the paper. Furthermore, the authors would like to thank Nepomuk Krenn (RICAM) and Yusuf Elbadry (CPS, TU Darmstadt) for their fruitful discussions.

Author Contributions Conceptualization: Ahmed Mohamed Jubartalla Ali, Peter Gangl, Mario Kapl; Methodology: Ahmed Mohamed Jubartalla Ali, Abdulmajeed Altassan, Théodore Cherrière, Peter Gangl, Mario Kapl; Software: Ahmed Mohamed Jubartalla Ali; Formal analysis: Théodore Cherrière, Peter Gangl, Margit Gföhler, Mario Kapl; Investigation: Ahmed Mohamed Jubartalla Ali, Abdulmajeed Altassan; Writing—Original Draft: Ahmed Mohamed Jubartalla Ali; Writing—Review & Editing: All authors; Visualization: Ahmed Mohamed Jubartalla Ali; Supervision: Margit Gföhler, Mario Kapl.

Funding Open access funding provided by Carinthia University of Applied Sciences (CUAS). The research presented in this paper has been supported by the Federal Ministry for Digital and Economic Affairs (BMDW) within the framework of COIN "Aufbau," 8th call of the Austrian Research Promotion Agency (FFG) - project number 884136 (iLEAD). P. Gangl and T. Cherrière are partially supported by the State of Upper Austria.

Declarations

Conflict of interest The authors have no conflict of interest to declare that are relevant to the content of this article.

Replication of results All information required to replicate the results of this work is given in the manuscript. Interested readers may contact the corresponding author for further clarification.

References

Ali AMJ (2025). Extending uni-variate interpolation schemes by the concept of Coons patches. https://doi.org/10.5281/zenodo.15017266

Ali AM, Faller LM, Gföhler M, Oswald F, Riemelmoser MK (2024) A two-scale topology optimization method for functionally graded lattice structures using three families of micro-structures. Comput Aided Des Appl 21(2):179–198. https://doi.org/10.14733/cadaps.2024.179-198

Andreassen E, Andreasen CS (2014) How to determine composite material properties using numerical homogenization. Comput Mater Sci 83:488–495. https://doi.org/10.1016/j.commatsci.2013.09.006

Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O (2010) Efficient topology optimization in Matlab using 88 lines of code. Struct Multidisc Optim 43(1):1–16. https://doi.org/10.1007/s00158-010-0594-7

Augusto O, Palma LF (2022) Some considerations on multi-material topology optimization using ordered SIMP. Struct Multidisc Optim. https://doi.org/10.1007/s00158-022-03379-7

Barnhill RE, Birkhoff G, Gordon WJ (1973) Smooth interpolation in triangles. J Approx Theory 8(2):114–128. https://doi.org/10.1016/0021-9045(73)90020-8

Bendsøe MP, Sigmund O (2004) Topology optimization. Theory, methods, and applications. Springer, Berlin. https://doi.org/10.1007/978-3-662-05086-6

Bendsøe MP (1989) Optimal shape design as a material distribution problem. Struct Optim 1(4):193–202. https://doi.org/10.1007/bf01650949

Bourdin B (2001) Filters in topology optimization. Int J Numer Methods Eng 50(9):2143–2158. https://doi.org/10.1002/nme.116

Bruyneel M (2011) SFP-a new parameterization based on shape functions for optimal material selection: application to conventional composite plies. Struct Multidisc Optim 43(1):17–27. https://doi.org/10.1007/s00158-010-0548-0

Cherrière T, Laurent L, Hlioui S, Louf F, Duysinx P, Geuzaine C, Ben Ahmed H, Gabsi M, Fernández E (2022) Multi-material topology optimization using Wachspress interpolations for designing a 3-phase electrical machine stator. Struct Multidisc Optim 65(12):352. https://doi.org/10.1007/s00158-022-03460-1

Coons SA (1967) Surfaces for computer-aided design of space forms. Technical report, USA. https://dl.acm.org/doi/book/10.5555/889976

Costa MR, Sohouli A, Suleman A (2022) Multi-scale and multi-material topology optimization of gradient lattice structures using surrogate models. Compos Struct 289:115402–115402. https://doi.org/10.1016/j.compstruct.2022.115402

Deetman A (2019) Python code of the method of moving asymptotes. https://github.com/arjendeetman/GCMMA-MMA-Python Accessed 12 March 2024

Deng Z, Lei Z, Cheng G, Liang Y (2024) Approach for multi-valued integer programming in multi-material topology optimization: random discrete steepest descent (RDSD) algorithm. Comput Methods Appl Mech Eng 432:117449–117449. https://doi.org/10.1016/j.cma.2024.117449

Dinh TD, Hedayatrasa S, Bormann F, Bosman M, Paepegem WV (2024) A smooth single-variable-based interpolation function for multi-material topology optimization. Eng Comput. https://doi.org/10.1007/s00366-024-01945-9

Gao T, Zhang W (2011) A mass constraint formulation for structural topology optimization with multiphase materials. Int J Numer Methods Eng 88(8):774–796. https://doi.org/10.1002/nme.3197

Gaynor AT, Meisel NA, Williams CB, Guest JK (2014) Multiple-material topology optimization of compliant mechanisms created via polyjet three-dimensional printing. J Manuf Sci Eng. https://doi.org/10.1115/1.4028439

Gibiansky LV, Sigmund O (2000) Multiphase composites with extremal bulk modulus. J Mech Phys Solids 48(3):461–498. https://doi.org/10.1016/s0022-5096(99)00043-5

Gu X, He S, Dong Y, Song T (2022) An improved ordered SIMP approach for multiscale concurrent topology optimization with multiple microstructures. Compos Struct 287:115363–115363. https://doi.org/10.1016/j.compstruct.2022.115363

Guo Y, Liu C, Jia Y, Shen C, Guo X (2024) A multi-material topology optimization method based on implicit topology description functions. Comput Methods Appl Mech Eng 436:117676–117676. https://doi.org/10.1016/j.cma.2024.117676

Hu J, Luo Y, Liu S (2021) Two-scale concurrent topology optimization method of hierarchical structures with self-connected multiple lattice-material domains. Compos Struct 272:114224–114224. https://doi.org/10.1016/j.compstruct.2021.114224

Huang X, Li W (2021) A new multi-material topology optimization algorithm and selection of candidate materials. Comput Methods Appl Mech Eng 386:114114. https://doi.org/10.1016/j.cma.2021.114114

Jiang J, Li Z, Hu Y, Chen S, Song Y, Hu L (2024) Density-based topology optimization of multi-condition peening pattern for laser peen forming. Int J Mech Sci 267:108968–108968. https://doi.org/10.1016/j.ijmecsci.2024.108968

Li H, Luo Z, Gao L, Qin Q (2017) Topology optimization for concurrent design of structures with multi-patch microstructures by level sets. Comput Methods Appl Mech Eng 331:536–561. https://doi.org/10.1016/j.cma.2017.11.033

Liang Y, Cheng G (2019) Topology optimization via sequential integer programming and canonical relaxation algorithm. Comput Methods Appl Mech Eng 348:64–96. https://doi.org/10.1016/j.cma.2018.10.050

Liang Y, Sun K, Cheng G (2020) Discrete variable topology optimization for compliant mechanism design via sequential approximate integer programming with trust region (SAIP-TR). Struct Multidisc Optim 62(6):2851–2879. https://doi.org/10.1007/s00158-020-02693-2

Liao H (2021) An incremental form interpolation model together with the Smolyak method for multi-material topology optimization. Appl Math Model 90:955–976. https://doi.org/10.1016/j.apm.2020.10.017

Liao H, Ding W, Ai S, Gao R (2024) A single variable stress-based multi-material topology optimization method with three-dimensional unstructured meshes. Comput Methods Appl Mech Eng 421:116774–116774. https://doi.org/10.1016/j.cma.2024.116774

Liu L, Yan J, Cheng G (2008) Optimum structure with homogeneous optimum truss-like material. Comput Struct 86(13–14):1417–1425. https://doi.org/10.1016/j.compstruc.2007.04.030

Liu B, Yan X, Li Y, Zhou S, Huang X (2021) Multi-material topology optimization of structures using an ordered ersatz material model. Comput Model Eng Sci 128(2):523–540. https://doi.org/10.32604/cmes.2021.017211

Liu H, Wang C, Zhang Y, Liang Y (2023) Multi-material structural discrete variable topology optimization with minimum length scale control under mass constraint. Comput Methods Appl Mech Eng 420:116701–116701. https://doi.org/10.1016/j.cma.2023.116701

Lund E, Stegmann J (2004) On structural optimization of composite shell structures using a discrete constitutive parametrization. Wind Energy 8(1):109–124. https://doi.org/10.1002/we.132

Nguyen M-N, Lee D (2024) Design of the multiphase material structures with mass, stiffness, stress, and dynamic criteria via a modified ordered SIMP topology optimization. Adv Eng Softw 189:103592–103592. https://doi.org/10.1016/j.advengsoft.2023.103592

Qian X, Sigmund O (2011) Isogeometric shape optimization of photonic crystals via Coons patches. Comput Methods Appl Mech Eng 200(25–28):2237–2255. https://doi.org/10.1016/j.cma.2011.03.007

Sanders ED, Aguiló MA, Paulino GH (2018a) Multi-material continuum topology optimization with arbitrary volume and mass constraints. Comput Methods Appl Mech Eng 340:798–823. https://doi.org/10.1016/j.cma.2018.01.032

Sanders ED, Pereira A, Aguiló MA, Paulino GH (2018b) Polymat: an efficient Matlab code for multi-material topology optimization. Struct Multidisc Optim 58(6):2727–2759. https://doi.org/10.1007/s00158-018-2094-0

Sigmund O (1997) On the design of compliant mechanisms using topology optimization. Mech Struct Mach 25(4):493–524. https://doi.org/10.1080/08905459708945415

Sigmund O (2007) Morphology-based black and white filters for topology optimization. Struct Multidisc Optim 33(4–5):401–424. https://doi.org/10.1007/s00158-006-0087-x

Sigmund O (2022) On benchmarking and good scientific practise in topology optimization. Struct Multidisc Optim. https://doi.org/10.1007/s00158-022-03427-2

Sivapuram R, Picelli R (2017) Topology optimization of binary structures using integer linear programming. Finite Elem Anal Des 139:49–61. https://doi.org/10.1016/j.finel.2017.10.006

Stegmann J, Lund E (2005) Discrete material optimization of general composite shell structures. Int J Numer Methods Eng 62(14):2009–2027. https://doi.org/10.1002/nme.1259

Taheri AH, Suresh K (2016) An isogeometric approach to topology optimization of multi-material and functionally graded structures. Int J Numer Methods Eng 109(5):668–696. https://doi.org/10.1002/nme.5303

Wan C, Jiao H, Lv L, Lu C (2024) Multi-material topology optimization based on multiple simp of variable density method. J Mech Sci Technol 38(2):749–759. https://doi.org/10.1007/s12206-024-0124-y

Wang C, Zhu JH, Zhang WH, Li SY, Kong J (2018) Concurrent topology optimization design of structures and non-uniform parameterized lattice microstructures. Struct Multidisc Optim 58(1):35–50. https://doi.org/10.1007/s00158-018-2009-0

Wang Y, Hu D, Wang H, Zhang T, Yan H (2020a) Practical design optimization of cellular structures for additive manufacturing. Eng Optim 52(11):1887–1902. https://doi.org/10.1080/0305215X.2019.1696785

Wang C, Gu X, Zhu J, Zhou H, Li S, Zhang W (2020b) Concurrent design of hierarchical structures with three-dimensional parameterized lattice microstructures for additive manufacturing. Struct Multidisc Optim 61(3):869–894. https://doi.org/10.1007/s00158-019-02408-2

Wang J, Zhu J, Liu T, Wang Y, Zhou H, Zhang W-H (2023) Topology optimization of gradient lattice structure under harmonic load based on multiscale finite element method. Struct Multidisc Optim. https://doi.org/10.1007/s00158-023-03652-3

Wu H, Zhang P, Lin G-H (2015) Smoothing approximations for some piecewise smooth functions. J Oper Res Soc China 3(3):317–329. https://doi.org/10.1007/s40305-015-0091-1

Xu S, Liu J, Zou B, Li Q, Ma Y (2021) Stress constrained multi-material topology optimization with the ordered SIMP method. Comput Methods Appl Mech Eng 373:113453–113453. https://doi.org/10.1016/j.cma.2020.113453

Yang Z, Zhang C, Liao W, Liu T, Yang H (2024) Efficient and exquisite concurrent optimization of hierarchical structures with non-uniform eccentric body centered cubic lattice. Comput Methods Appl Mech Eng 423:116862. https://doi.org/10.1016/j.cma.2024.116862

Yi B, Yoon GH, Zheng R, Liu L, Li D, Peng X (2023) A unified material interpolation for topology optimization of multi-materials. Comput Struct 282:107041–107041. https://doi.org/10.1016/j.compstruc.2023.107041

Yin L, Ananthasuresh GK (2001) Topology optimization of compliant mechanisms with multiple materials using a peak function material interpolation scheme. Struct Multidisc Optim 23(1):49–62. https://doi.org/10.1007/s00158-001-0165-z

Zhang Y, Xiao M, Li H, Gao L, Chu S (2018) Multiscale concurrent topology optimization for cellular structures with multiple microstructures based on ordered SIMP interpolation. Comput Mater Sci 155:74–91. https://doi.org/10.1016/j.commatsci.2018.08.030

Zhang C, Xu S, Liu J, Ma Y (2022) Comprehensive clustering-based topology optimization for connectable multi-scale additive manufacturing structures. Addit Manuf 54:102786–102786. https://doi.org/10.1016/j.addma.2022.102786

Zhang C, Wu T, Xu S, Liu J (2023) Multiscale topology optimization for solid-lattice-void hybrid structures through an ordered multi-phase interpolation. Comput Aided Des 154:103424–103424. https://doi.org/10.1016/j.cad.2022.103424

Zhou M, Rozvany GIN (1991) The COC algorithm, Part II: topological, geometrical and generalized shape optimization. Comput Methods Appl Mech Eng 89(1–3):309–336. https://doi.org/10.1016/0045-7825(91)90046-9

Zhou H, Zhu J, Wang C, Zhang Y, Wang J, Zhang W (2022) Hierarchical structure optimization with parameterized lattice and multiscale finite element method. Struct Multidisc Optim. https://doi.org/10.1007/s00158-021-03149-x

Zuo W, Saitou K (2016) Multi-material topology optimization using ordered SIMP interpolation. Struct Multidisc Optim 55(2):477–491. https://doi.org/10.1007/s00158-016-1513-3