

A new three-level mesh method to accelerate the structural topology optimization



Yingjun Wang^a, Wei Zheng^a, Yongfeng Zheng^{a,b,*}, Daicong Da^c

^a National Engineering Research Center of Novel Equipment for Polymer Processing, The Key Laboratory of Polymer Processing Engineering of the Ministry of Education (South China University of Technology), Guangdong Provincial Key Laboratory of Technique and Equipment for Macromolecular Advanced Manufacturing, South China University of Technology, Guangzhou 510641, China

^b Department of Civil Engineering, University of Siegen, Siegen D-57076, Germany

^c Department of Mechanical Engineering, Northwestern University, Evanston, IL 60208, USA

ARTICLE INFO

Article history:

Received 29 November 2021

Revised 11 April 2022

Accepted 6 May 2022

Available online 8 May 2022

Keywords:

Topology optimization
Adaptive three-level mesh
Serendipity elements
Finite element method

ABSTRACT

To accelerate the structural topology optimization, this paper proposes an adaptive three-level mesh method (ATMM) to reduce the computational costs without loss of accuracy. The ATMM divides the elements into three levels: fine elements, middle elements and coarse elements. Topology optimization is initially performed on the uniform fine meshes, when adjacent fine elements change into fully voids or solids during optimization, they will merge into middle elements, and such merging processes are the same as those between middle elements and coarse elements. Meanwhile, when merged middle elements or coarse elements become gray, they will return to the fine elements. To handle the incompatibility of adjacent elements in different levels, the hybrid-order serendipity elements are adopted. This paper proposes a new nodal numbering scheme to assemble the global stiffness matrix, and a new sensitivity filter scheme is discussed to avoid the numerical instability. Additionally, ATMM can obtain better acceleration and convergence, when combining with the existing gray-scale suppression technology. Lastly, four examples are provided to verify the proposed method, optimization results are consistent with those obtained from the uniform fine meshes, but with greatly reduced computational costs. In the four numerical examples, the time consumed of ATMM in each iteration is only about 30%~69% compared to that of the uniform fine mesh, and the total iteration numbers can be reduced by 18%~62%.

© 2022 Elsevier Inc. All rights reserved.

1. Introduction

Topology optimization is to find the best material distribution within a given design domain under the prescribed constraints, with the purpose of pursuing the best structural performance. Since the seminal work of homogenization method proposed by Bendsøe and Kikuchi [1], topology optimization has achieved vigorous developments due to its advantages in flexible and creative design. Over the past decades, many remarkable topology optimization methods have been proposed, such as the solid isotropic material with penalization (SIMP) method [2,3], the evolutionary structural optimization (ESO)

* Corresponding author at: National Engineering Research Center of Novel Equipment for Polymer Processing, The Key Laboratory of Polymer Processing Engineering of the Ministry of Education (South China University of Technology), Guangdong Provincial Key Laboratory of Technique and Equipment for Macromolecular Advanced Manufacturing, South China University of Technology, Guangzhou 510641, China

E-mail address: hnzyf@scut.edu.cn (Y. Zheng).

and bi-directional ESO methods [4], the level-set method (LSM) [5], the explicit moving morphable components (MMC) method [6], and some non-gradient topology optimization methods [7,8]. Topology optimization was initially utilized to solve the minimum compliance problem in solid mechanics, but so far, it has been extended to be applied in many interesting fields such as heat transfer [9], electromagnetics [10], optics [11] and acoustics [12]. Additionally, some attracting research directions have been developed such as stress-constrained topology optimization [13], isogeometric topology optimization [14,15], topology optimization for additive manufacturing [16], topology optimization considering uncertainty [17] and machining learning-based topology optimization [18].

In general, most topology optimization problems are solved based on gradient information, which requires the sensitivities with respect to the objective function and constraints. As the topology optimization models become more and more complex with large number of design variables and degree of freedoms (DOFs), the efficiency cannot meet the practical requirements, especially for the large-scale engineering structures. Therefore, a high-efficient computational method without loss of accuracy is still important and essential.

To reduce the computational costs of topology optimization, many researchers have made quite a few extraordinary works. In terms of improving the efficiency of finite element analysis (FEA), one of the most time-consuming part in topology optimization, Wang et al. [19] improved the solving efficiency of finite element (FE) equations by recycling the search spaces from previous linear systems, and the minimum residual method is used to make the recycle more efficient. Amir et al. [20] introduced a reanalysis technology to approximate solution of FE equations with low computational cost; in addition to this, Amir et al. [21] also speeded the solution of large-scale FE equations in nested analysis problems, by using multigrid preconditioned conjugate gradients (MGCG) solver. Gogu [22] constructed basis model by employing previous several solutions of FE equations to construct reduced basis model, so as to improve the solving efficiency. The XFEM method is efficient to perform FEA in the level set method, which can get rid of the void areas and improve the boundary accuracy without adding additional DOF [23,24]. Liao et al. [25] proposed an initial-value-based preconditioned conjugate-gradient (PCG) method to reduce the iterations solving FE equations. Zheng et al. [26] reduced the DOFs of FE equations by avoiding calculating the DOFs of void nodes. In addition, several works are concentrated on reducing the iteration numbers to accelerate the topology optimization process. Guest and Smith Genut [27] reduced the dimension of design variables by separating the design mesh from the analysis mesh, which leads to fewer iterations of optimization process. Kim et al. [28] reduced the quick converged design variables to accelerate the convergence of the topology optimization. Wang et al. [29] proposed a local-update strategy to improve the convergence performance of isogeometric topology optimization. Li et al. [30] recommended applying Anderson extrapolations periodically in the fixed-point iteration to improve the convergence performance. Recently, the neural networks have been developed to accelerate the topology optimization [31,32]. Through neural networks, one can directly obtain the final optimized topology without the time-consuming FEA and redundant iterations. However, for those problems with large-scale models, complex boundary conditions and irregular meshes, neural networks for topology optimization currently have some limits to be further overcome. Parallel computing technologies have also been applied to the topology optimization to improve the computational efficiency [33,34]. However, it is implemented with the help of the computational ability of the powerful hardware, rather than the improvement of optimization algorithms themselves.

Apart from the above-mentioned accelerating methods, a series of methods involving so-called adaptive mesh refinement (AMR) has also been developed for obtaining lower computational costs and preserves relatively accurate topology layout. Conventional topology optimization methods are mostly performed on the fixed uniform meshes, with a much large number of elements to acquire accurate optimized results, while AMR dedicates to obtain results with the same accuracy as those obtained from uniform fine elements, but with lower computational costs by adaptively placing elements with different sizes to reduce the total elemental numbers. AMR not only improves the solving efficiency of FE equations, but also reduces the computational costs of sensitivity analysis and the update processes of design variables, which has been one of the most popular methods for accelerating topology optimization. Maute and Ramm [35] proposed an adaptive topology optimization method for 2D maximum stiffness problem, where elemental size and orientation in analysis model were adaptively changed with the material distribution in design model, and the void domain in analysis model was neglected. DeRose et al. [36] used the octree data structure to refine the areas where density distribution varies rapidly. Lin and Chou [37] proposed a two-stage method for homogenization topology optimization, which used large size of elements in the first stage for the initial topology structure and detailed it in the second stage by using sufficiently fine elements. Costa and Alves [38] employed h -adaptive refinement in topology optimization, in which the elements with density bigger than 0.5 and boundary elements were refined. Stainko [39] only refined the boundary through multilevel mesh approach. Nguyen et al. [40] proposed a multiresolution topology optimization method (MTOP), where three different kinds of meshes with different functions were employed: the displacement mesh for FEA, the design variable mesh for optimization, and the density mesh for material distribution. The design variable mesh and density mesh were fine discretized to obtain relatively high-resolution topological structure, but the displacement mesh was coarsely discretized so that the FEA efficiency was improved. Lieu and Lee introduced the MTOP scheme in the framework of isogeometric analysis (IGA) to reduce the computational cost of topology optimization of IGA with high resolution optimized results [41,42]. Bruggi and Verani [43] adaptively changed the underlying grid with triangle elements after each optimization cycle based on the indicators of compliance error and topological error. Wang et al. [44] used an adaptive design variable point to describe material density filed by Shepard interpolants to improve interface resolution, and the FEA mesh was coarsely fixed to save computational cost. Nana et al. [45] applied h -adaptation to refine interface area, and coarsened fully solid and void areas for tetrahedral meshes. Chau

et al. [46] proposed an adaptive polygonal mesh for multi-material topology optimization based on polytree data structure. Salazar de Troya and Tortorelli [47] developed an adaptive mesh refinement in stress-constrained topology optimization. Vogel and Junker [48] presented an adaptive thermodynamic topology optimization method, and the adaptability was also utilized to enlarge coarse structures for more detailed structures.

However, there are still some disadvantages in the existing AMR methods such as: (a) The FEA meshes are mostly refined every certain number of iterations, i.e., the FEA meshes are updated in stages. Thus, the FEA meshes are not totally adaptive to the topological changes of structures, which decreases the FEA accuracies. (b) Since elements in different sizes appear on a mesh, the so-called hanging nodes may emerge. To overcome the incompatibility between adjacent elements, a type of widely used method adds constraints on the DOFs of the hanging nodes through Lagrange multipliers or penalization [48–50]. Since the DOFs of hanging nodes are enforced to satisfy the supposed constraints, which implies no DOFs exist at the hanging nodes, the FEA accuracies of the elements with hanging nodes are compromised [51]. (c) Existing AMR methods start with coarse meshes and then refine locally or globally. From the perspective of optimization, the coarse meshes in the early stage will make the search space become smaller compared to fine meshes, which makes it easy to fall into local optimal solutions. Although subsequent refinement processes are performed, the final designs have non-neglected errors induced by the results of coarse mesh stage. (d) For the AMR-related methods that separates design variable meshes from FEA meshes, the design variable meshes are fine and adaptive to obtain high-resolution topological solutions, while FEA meshes are coarsely fixed to decrease the computational costs. Although the design variable meshes are fine enough, coarse FEA meshes may result in inaccurate optimization results.

To overcome the abovementioned disadvantages, this study proposes an adaptive three-level mesh method (ATMM) with the following advantages: (a) ATMM updates the mesh in each iteration, once the lower-level elements together change into voids or solids, they will merge into higher-level elements, and once the higher-level elements change into gray elements, they will return to the fine elements. Thus, the constructed meshes of ATMM are always conformed to the topological structure in each iteration. (b) ATMM employs the hybrid-order serendipity elements to construct shape functions of elements with hanging nodes, which satisfies the shape function interpolation conditions as the standard finite elements and the DOFs of hanging nodes are independently calculated instead of being dominated by other DOFs as the conventional AMR methods, so as to guarantee the solution accuracy. (c) ATMM starts from the uniform fine elements ensuring that there is a large enough search space, and then coarsens the fully solid or void areas where the design variables tend to be stable, thus the suboptimal results potentially caused by the coarse meshes in the early stage can be avoided. (d) ATMM adaptively places higher-level elements in solid or void regions to construct the FEA meshes, which improves the FEA accuracies of the topological structures compared to those obtained from the fixed coarse meshes.

The remainder of this paper is organized as follows. Section 2 presents a brief overview of SIMP-based topology optimization. Section 3 discusses the ATMM including the three-level meshing scheme for design domain, the FEA for the three-level mesh, and the implementation of sensitivity analysis and filter. The flowchart of the algorithm implementation is presented in Section 4. Numerical examples including MBB beam, 3D cantilever, and compliance mechanism are shown in Section 5, and conclusions are given in Section 6.

2. An overview of SIMP-based topology optimization

In SIMP-based topology optimization, the material distribution problem of the topology optimization is modeled as material density distribution. Each element in design domain is assigned a density between 0 and 1, which determines elemental Young's modulus as follows:

$$E_e(x_e) = E_{\min} + x_e^p(E_0 - E_{\min}), \quad x_e \in [0, 1] \tag{1}$$

where x_e is the e th elemental density and E_e is its Young's modulus, E_0 is the Young's modulus of solid elements whose densities are equal to 1. E_{\min} is the Young's modulus for void elements, which is a small value to avoid the singularity of stiffness matrix, and p is the penalization factor to push the element densities toward 0 or 1.

Compliance minimization subjecting to a given volume is the classical problem in topology optimization, which can be formulated as

$$\begin{aligned} \text{Find : } & x = (x_1, x_2, x_3, \dots, x_N)^T \\ \text{Min : } & c(x) = \frac{1}{2}U^TKU = \frac{1}{2} \sum_{e=1}^N E_e(x_e)u_e^T k_0 u_e \\ \text{s.t. : } & \begin{cases} V(x) \leq V_0 f \\ KU = F \\ 0 \leq x \leq 1 \end{cases} \end{aligned} \tag{2}$$

where $x=(x_1, x_2, x_3, \dots, x_N)^T$ is the design variable vector; N is the total number of elements; c is the structural compliance; U and u_e are the global displacement vector and element displacement vector, respectively; k_0 is the element stiffness matrix for elements with unit Young's modulus; K is the global stiffness matrix; V_0 and $V(x)$ are the volume of the design domain and material usages, respectively; f is the prescribed volume fraction.

The above optimization problem can be solved by using either gradient-based algorithms such as optimality criteria (OC), method of moving asymptotes (MMA) and sequential quadratic programming (SQP) or non-gradient methods such as genetic

algorithms (GA), detailed information about these algorithms can be referred to [52,53]. This paper adopts OC method as the optimization solver due to its numerical efficiency and simplicity, and a heuristic updating scheme of design variables according to OC is given by

$$x_e^{new} = \begin{cases} \max(0, x_e - m) & \text{if } x_e B_e^\eta \leq \max(0, x_e - m) \\ \min(1, x_e + m) & \text{if } x_e B_e^\eta \geq \min(1, x_e + m) \\ x_e B_e^\eta & \text{otherwise} \end{cases} \tag{3}$$

where m is a positive move limit, η is a numerical damping coefficient, B_e is obtained from the optimality condition as

$$B_e = -\frac{\frac{\partial c}{\partial x_e}}{\lambda \frac{\partial V}{\partial x_e}} \tag{4}$$

where Lagrange multiplier λ must satisfy the volume constraint, i.e., $V(x(\lambda)) = V_{of}$, and it is iteratively calculated by the bisection algorithm.

Sensitivities of the objective function c and the material volume V with respect to the elemental density x_e are derived as

$$\frac{\partial c}{\partial x_e} = -\frac{1}{2} p x_e^{p-1} (E_0 - E_{min}) u_e^T k_e u_e \tag{5}$$

$$\frac{\partial V}{\partial x_e} = V_e \tag{6}$$

where V_e is the volume of the e th element.

To address the numerical problems in topology optimizations such as checkboard patterns, mesh-dependency and local minima, the sensitivity filter scheme is utilized in this paper. The sensitivity is modified as

$$\widehat{\frac{\partial c}{\partial x_e}} = \frac{1}{\max(\gamma, x_e) \sum_{i \in N_e} H_{ei}} \sum_{i \in N_e} H_{ei} x_i \frac{\partial c}{\partial x_i} \tag{7}$$

where γ is a small positive number (typically $\gamma=0.001$) introduced to avoid division by 0, and N_e is the set of elements i defined as

$$N_e = \{i \mid \text{dist}(i, e) \leq r_{min}\} \tag{8}$$

where $\text{dist}(i, e)$ represents the distance between the centers of element e and element i , and r_{min} is the filter radius. The weight factor H_{ei} in Eq. (7) can be written as

$$H_{ei} = \max(0, r_{min} - \text{dist}(i, e)) \tag{9}$$

3. An adaptive three-level meshing method

3.1. Three-level meshing scheme for design domain

In density-based topology optimization, it is not necessary to apply over-dense mesh to the purely solid or void regions of a structure [44]. Instead of using globally fixed mesh, ATMM places coarse meshes in the purely solid or void regions, and places fine meshes in the gray regions, which are the transition regions between solid and void regions, to improve the computational efficiency without loss of accuracy.

To implement the three-level meshing scheme, the elements are divided into three different levels named fine elements (level 1), middle elements (level 2) and coarse elements (level 3), respectively, as shown in Fig. 1 (2D problem). The design domain is divided by fine elements initially, as optimization progresses, some elements will become solids or voids. When four fine elements together become solids or voids in one iteration, they will merge into a middle element, and when four middle elements together become solids or voids, they will merge into a coarse element.

It is worth noting that only the two elements from the same level or adjacent levels are allowed to be adjacent to each other, i.e., coarse elements (level 3) cannot be adjacent to the fine elements (level 1). If coarse elements are allowed to be adjacent to fine elements as shown in Fig. 2, in order to preserve the compatibility between the elements in level 3 and level 1, the number of nodes on each element boundary may be 3 or 2 (see Fig. 2). Besides, it may also be 1 in the case that elements are adjacent in adjacent levels. The diversity of the number of boundary nodes increases the complexity of programming and reduces the overall computational efficiency. Therefore, the coarse elements cannot be adjacent to the fine elements, which is considered as a constraint in the three-level meshing scheme.

To identify whether four fine elements together can merge into a middle element, as shown in Fig. 3 (a), fine elemental densities are projected to the densities of the center nodes of middle elements, which are termed as M-nodes herein. The projection criteria between the densities of fine elements and M-nodes is given by

$$denM_i = \frac{\sum_{e \in N_M} x_e}{4} \tag{10}$$

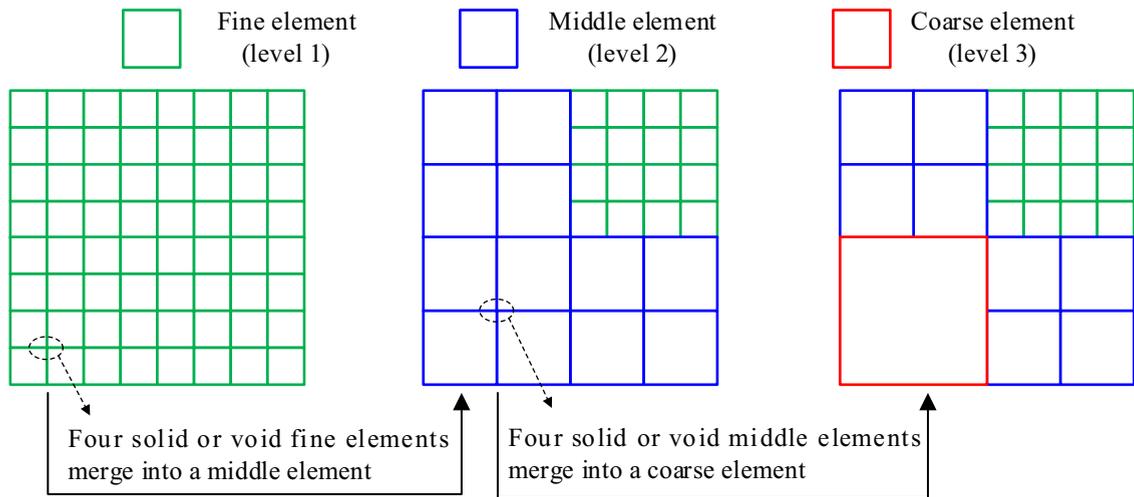


Fig. 1. The formation process of the three-level mesh.

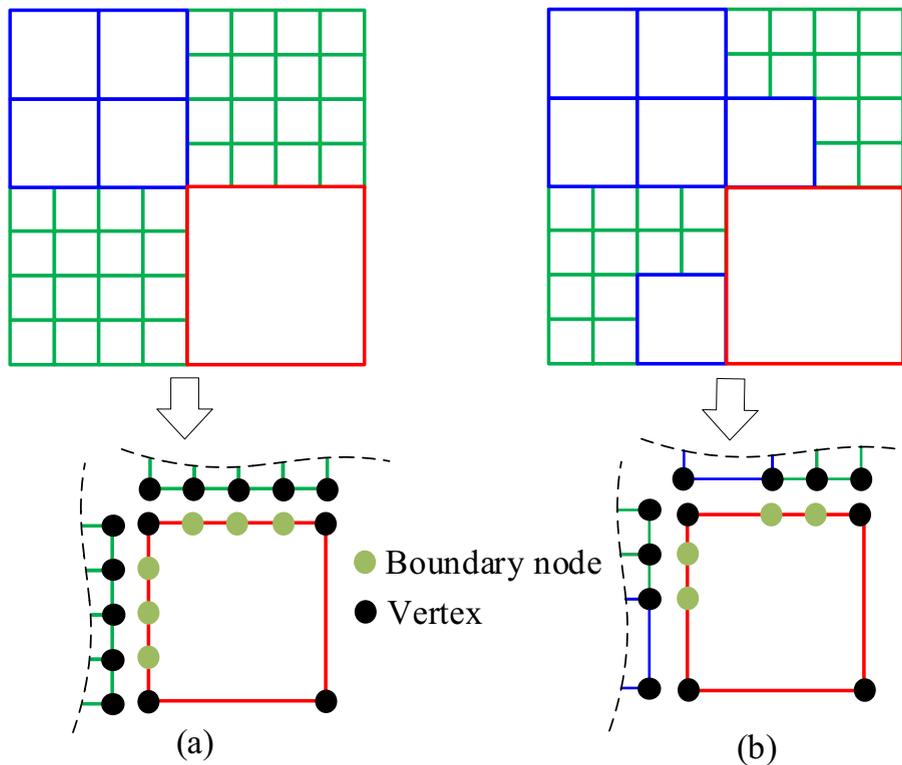


Fig. 2. Transition between the coarse and fine elements: (a) a coarse element is adjacent to fine elements, and (b) a coarse element is adjacent to both fine and middle elements.

where $denM_i$ represents the density of the i th M-node, N_M is the set of the four fine elements around i th M-node. After the projection, $denM_i = 0$ or 1 means the four elements around the i th M-node are all solid or void elements and these elements can merge into a middle element. Setting the M-nodes with 0 or 1 density as active M-nodes, the four elements around one active M-node can merge into a middle element.

To identify whether four middle elements together can merge into a coarse element, as shown in Fig. 3 (b), the densities of the M-nodes are projected to the densities of the center nodes of coarse element, which are termed as C-nodes hereafter.

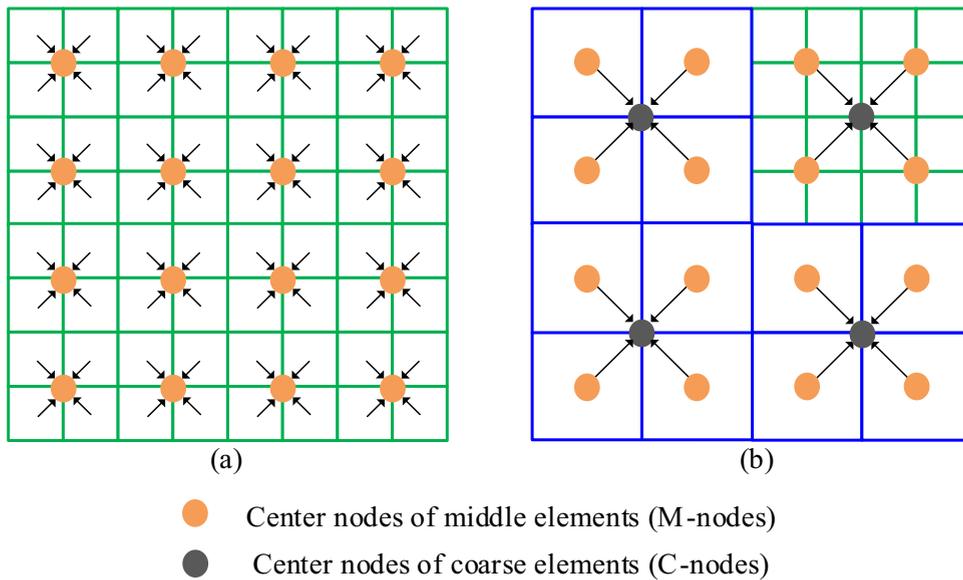


Fig. 3. Projecting element densities:(a) fine element densities to the densities of M-nodes, and (b) the densities of M-node to the densities of C-nodes.

The projection between the densities of M-nodes and C-nodes is given by

$$denC_j = \frac{\sum_{i \in N_c} denM_i}{4} \tag{11}$$

where $denC_j$ denotes the density of the j th C-node, N_c is the set of the four M-nodes around the j th C-node. After projection, four middle elements together around the j th C-node are void or solid can be parameterized by $denC_j = 0$ or 1. As mentioned above, four solid or void fine elements around one M-node can merge into a middle element. If the merged coarse element is adjacent to fine elements, four solid or void middle elements around one C-node are not permitted to merge. Therefore, $denC_j = 0$ or 1 is not the criterion that the four middle elements together around the j th C-node can merge into a coarse element, and some constraints must be added to avoid coarse and fine elements are adjacent. This can be implemented by checking the adjacent C-node densities around one C-node with 0 or 1 density. If the adjacent C-node densities are not all equal to 0 or 1, according to the two projection rules mentioned above, there are fine elements (level 1) adjacent to the coarse elements (level 3). For example, as shown in Fig. 4, the adjacent C-node densities around the 5th C-node (i.e., the densities of 6th, 8th, 4th and 2-nd C-nodes) are not all equal to 0 or 1 (the density of the 8th element is not equal to 0 or 1), and the fine elements will be adjacent to the coarse element if the 4 middle elements around 5th C-node emerged. It is obvious that if the density of a C-node and its adjacent C-node densities are all equal to 0 or 1, four middle elements around the C-node can merge into a coarse element. Setting the C-nodes satisfying the above two constraints as active C-nodes, four middle elements around the active C-nodes can merge into a coarse element.

If some C-node's adjacent C-nodes do not exist in the design domain, it is proper to check the densities of C-nodes within the design domain (e.g., as shown in Fig. 4, the 6th C-node does not have a downside C-node in the design domain, we can just check whether the densities of 3th, 5th and 9th C-nodes are all 0 or 1 to identify whether the four elements around the 6th node can merge into a coarse element).

In summary, by projecting fine element densities to M-node's densities and projecting M-node's densities to C-node's densities, four fine elements around an active M-node can merge into a middle element, and four middle elements around an active C-node can merge into a coarse element. It is noted that whether the C-nodes and M-nodes are active should be identified in each iteration, if one C-node or M-node is inactive, the merged middle elements will return to the fine elements, and the merged coarse elements will return to the middle elements or fine elements. Thus, the transformations between low-level elements and high-level elements are mutual.

The proposed three-level meshing scheme can be easily extended to 3D cases as shown in Fig. 5, and higher level elements can also be extended, i.e., coarse elements (level 3) can also merge into a coarser element (higher level) if it is necessary in large-scale problems.

In the three-level meshing scheme, if the initial fine element size is relatively big, the optimization accuracy may be influenced. The energy error indicator of strain field can be used to determine the initial fine element size, and ensure the accuracy when the elements are merged [54].

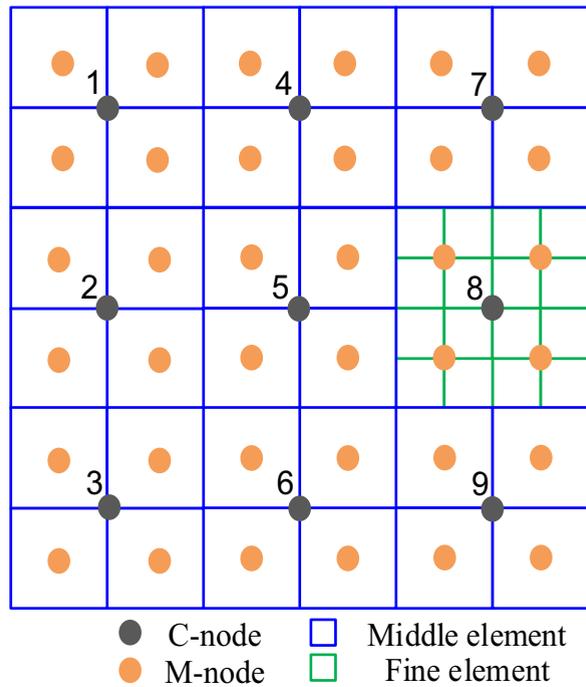


Fig. 4. Illustration whether four middle elements around a C-node can merge into a coarse element.

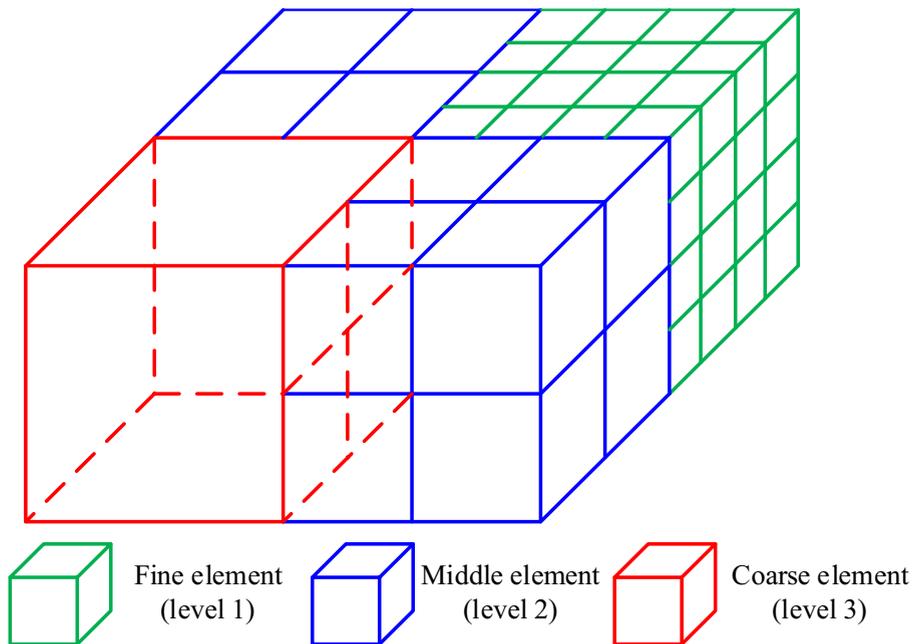


Fig. 5. The three-level meshing scheme for 3D case.

3.2. Finite element analysis for the three-level mesh

After the design domain is meshed according to the above-mentioned three-level meshing scheme, how to implement FEA is the key issue since the elements in different levels have different sizes. To preserve the compatibility of adjacent elements, the hybrid-order serendipity elements are introduced in this section, and the calculation of the element stiffness matrices for different types of serendipity elements and assembly of global stiffness matrix are also discussed.

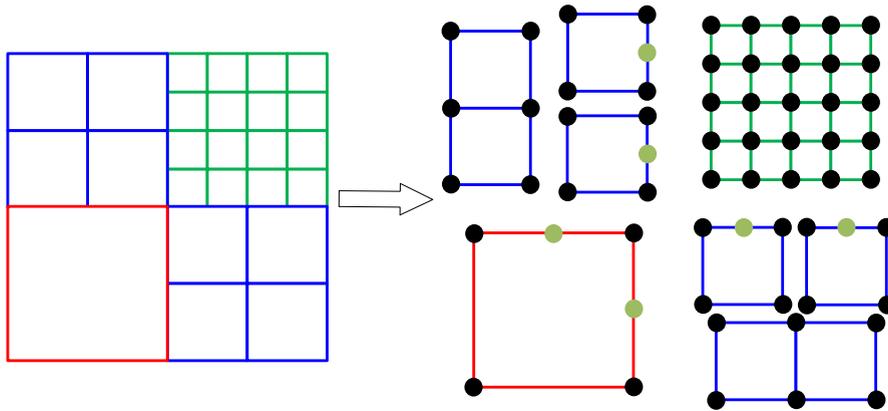


Fig. 6. Comparison of nodal distribution between Lagrange elements and serendipity elements in different orders. (a) Lagrange elements; (b) serendipity elements.

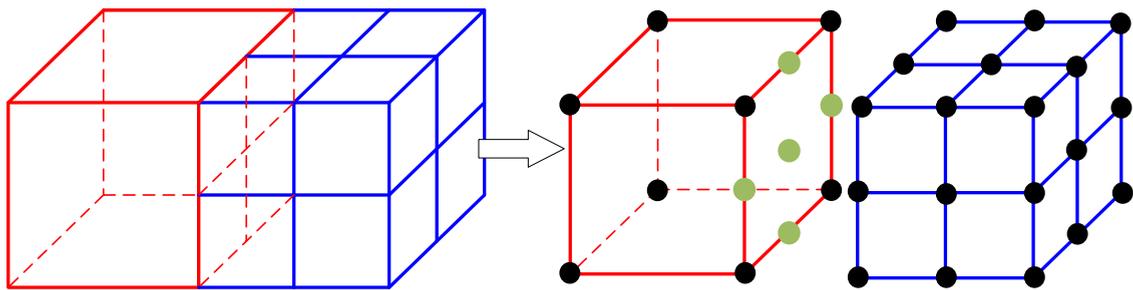


Fig. 7. The case that 5-node Serendipity element is utilized to preserve the compatibility between the elements of level 3 and level 2.

3.2.1. Hybrid-order serendipity elements

Compared to the conventional Lagrange elements whose nodes may exit inside elements, a major difference in serendipity elements is that they are constructed with nodes only on elemental boundaries, as shown in Fig. 6. For linear elements, the serendipity elements are identical to Lagrange elements. As the element order increases, the Lagrange elements have increasing interior nodes compared to serendipity elements without interior nodes, which will increase the number of DOFs. Due to the high efficiency, serendipity elements are used in this study, which will be further extended to the hybrid-order Serendipity elements that are the hybrid from linear and quadratic Serendipity elements. Cubic or higher order elements are out of our scope, their details can be referred to [55].

Another advantage of serendipity elements is that its availability in the transition between the elements with different sizes, since the boundary nodes (exclude vertices of an element) of serendipity elements can change, i.e., the nodes on the elemental boundaries may be present or absent. Thus, serendipity elements with 8-node and 4-node can produce those with 7-node, 6-node, and 5-node, which are named as hybrid-order serendipity elements which mix liner elements and quadratic elements. Although the hybrid-order serendipity elements employed in this paper are not standard serendipity elements, they are still regarded as general serendipity elements for convenience. As shown in Fig. 7, the 5-node serendipity element can be utilized to implement the compatibility between elements of level 3 and level 2. In Fig. 7, elements in level 2 exist on the right of level 3, but during optimization, elements in level 2 may exist in any orientations such as the down, right, up, and left of level 3. Therefore, for the compatibility between the elements in level 2 and level 3, elemental boundary nodes should also exist on any elemental boundaries in level 3. Thus, there are totally 16 (i.e., $\sum_{i=0}^4 C_4^i = 16$) types of serendipity elements in level 3 for the compatibility between the elements in level 2 and level 3, they are shown in Fig. 8.

The above discussion is about the compatibility between the elements in level 3 and level 2, and it is also suitable for the transition between the elements in level 2 and level 1. Hence, there are also 16 types serendipity elements in level 2, which are the same as the serendipity elements in level 3 shown in Fig. 8 except the different element sizes. Elements in level 1 do not need to be compatible with the element in the lower level, as it is the lowest level and has only one type of element: 4-node serendipity element.

Finally, the compatibility among three-level elements includes the compatibility between elements in level 3 and level 2, and the compatibility between the elements in level 2 and level 1, as shown in Fig. 9.

In 3D cases, the hexahedral serendipity elements can ensure the compatibility of the hexahedral elements in different levels, as shown in Fig. 10. Similar to the situation that nodes may exist on any boundary of rectangle elements in 2D

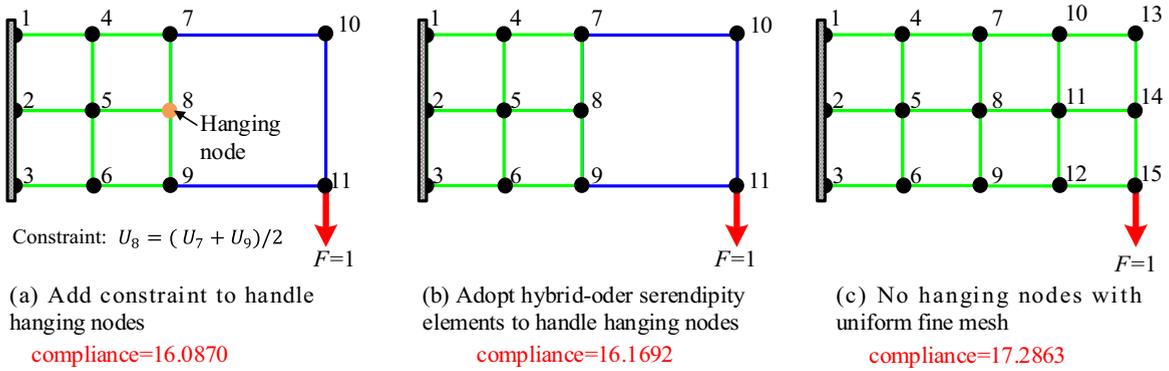


Fig. 8. The 16 types of serendipity elements in level 3.

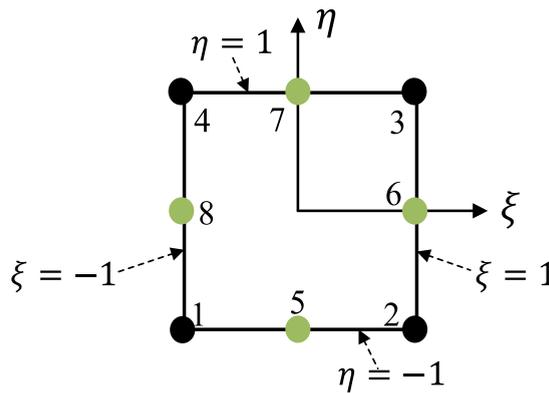


Fig. 9. Compatibility among the three-level elements.

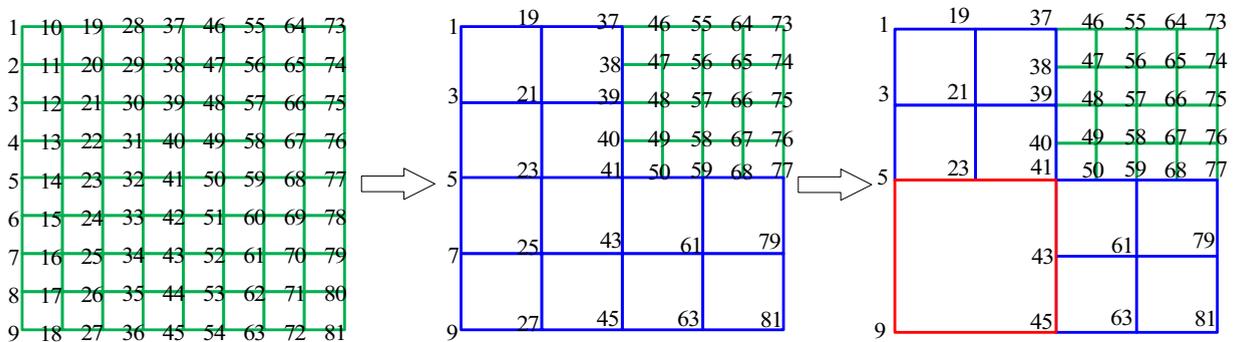


Fig. 10. Compatibility between the 3D elements in different levels.

cases, they may exist on any face of hexahedral elements in 3D cases; thus, there are totally 64 (i.e., $\sum_{i=0}^6 C_6^i = 64$) types of hexahedral serendipity elements for the compatibility between the hexahedral elements in different levels.

At the end of this section, we use a small test to discuss the advantages of adopting the hybrid-order serendipity elements to deal with hanging nodes with respect to FE accuracy compared to that of adding constraints in conventional AMR methods as mentioned in the introduction. The problem is to calculate the compliance of the cantilever beam as shown in Fig. 11. The Young's modulus and the Poisson's ratio of the cantilever beam are set to 1 and 0.3, respectively. Fig. 11 (a) adds constraint $U_8 = (U_7 + U_9)/2$ on the hanging node, Fig. 11 (b) adopts the abovementioned 5-node serendipity element, and Fig. 11 (c) is the benchmark with uniform fine mesh. Compared to the compliance (16.0870) in Fig. 11 (a), the compliance (16.1692) in Fig. 11 (b) is closer to the referenced compliance (17.2863) in Fig. 11 (c), which demonstrates the FE accuracy using hybrid-order serendipity elements is better than adding constraints on hanging nodes. It is without loss of generality

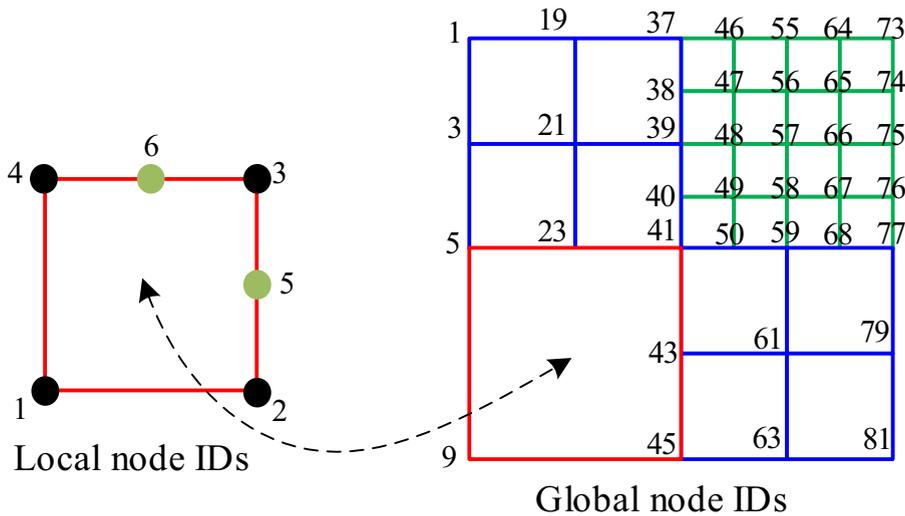


Fig. 11. The FE accuracy comparison. (a) Adding constraints to handle hanging nodes. (b) adopting hybrid-order serendipity elements to handle hanging nodes. (c) no hanging nodes with uniform fine mesh.

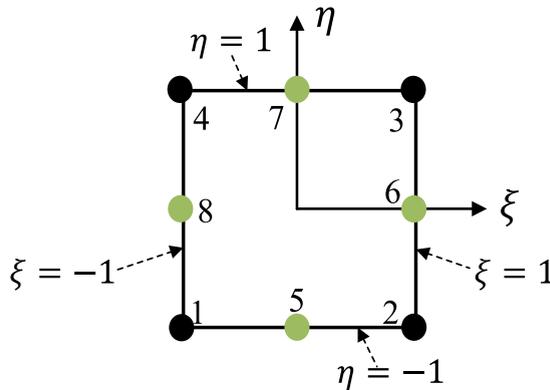


Fig. 12. Illustration for discussing the shape functions of each type of element.

since DOFs of hanging nodes are independently calculated in hybrid-order serendipity elements, while there is actually no DOFs existing on hanging nodes in the constraint-adding method. The hybrid-order serendipity elements have more independent DOFs than constraint-adding methods under the same mesh, so they have better performance in FE accuracy.

3.2.2. Calculation of the element stiffness matrices

To calculate the element stiffness matrices, shape functions of each type of element must be known in advance. Detailed derivation of shape functions can be referred to [56]. For the 8-node serendipity element shown in Fig. 12, its shape functions are given by

$$\begin{aligned}
 N_5 &= \frac{1}{2}(1 - \xi^2)(1 - \eta), & N_6 &= \frac{1}{2}(1 - \eta^2)(1 + \xi) \\
 N_7 &= \frac{1}{2}(1 - \xi^2)(1 + \eta), & N_8 &= \frac{1}{2}(1 - \eta^2)(1 - \xi) \\
 N_1 &= N'_1 - \frac{1}{2}N_5 - \frac{1}{2}N_8, & N_2 &= N'_2 - \frac{1}{2}N_5 - \frac{1}{2}N_6 \\
 N_3 &= N'_3 - \frac{1}{2}N_6 - \frac{1}{2}N_7, & N_4 &= N'_4 - \frac{1}{2}N_7 - \frac{1}{2}N_8
 \end{aligned}
 \tag{12}$$

where N_i ($i = 1, 2 \dots 8$) is the shape function at the i th node, and N'_j ($j = 1, 2, 3, 4$) is the shape function of well-known 4-node rectangle Lagrange element, which can be written as

$$N'_j = \frac{1}{4}(1 + \xi_j \xi)(1 + \eta_j \eta) \quad j = 1, 2, 3, 4
 \tag{13}$$

where (ξ_j, η_j) is the nature coordinate of node j . Up to now, the shape functions of 8-node serendipity elements are obtained. For the other 15 types of serendipity elements mentioned in Fig. 8, the following shape function construction method is utilized. If any node in 5, 6, 7 and 8 shown in Fig. 12 is absent, the shape function at that node is set to 0; for example,

Table 1
Identification criterion of different types in serendipity elements.

Element levels	Element types	Identification criterion
Middle element	4-node	All adjacent M-nodes around the element are active M-nodes
	5-node	Three of adjacent M-nodes around the element are active M-nodes
	6-node	Two of adjacent M-nodes around the element are active M-nodes
	7-node	One of adjacent M-nodes around the element is active M-nodes
	8-node	None of adjacent M-nodes around the element is active M-nodes
Coarse element	4-node	All adjacent C-nodes around the element are active C-nodes
	5-node	Three of adjacent C-nodes around the element are active C-nodes
	6-node	Two of adjacent C-nodes around the element are active C-nodes
	7-node	One of adjacent C-nodes around the element is active C-node
	8-node	None of adjacent C-nodes around the element is active C-node

if node 5 does not exist, we should set N_5 in Eq. (12) to 0, then the shape functions of the element excluding node 5 are constructed. In this way, the shape functions of other 15 types of serendipity elements can be obtained. The shape function construction of the 64 types in 3D serendipity elements is similar to the 2D case, which is generally implemented by simple interpolation using the shape function properties. The element stiffness matrix k_0 is the integral of the strain-displacement matrix B and elasticity matrix D in the form of

$$k_0 = \int_{\Omega} B^T D B d\Omega \tag{14}$$

where Ω denotes the design domain, D is a constant matrix determined by Young’s modulus and Poisson’s ratio, and B can be expressed as the differentiation of shape functions in the form of

$$B = [B_1 \ B_2 \ B_3 \ \dots \ B_n]$$

$$B_i = \begin{bmatrix} \partial N_i / \partial x & 0 & 0 \\ 0 & \partial N_i / \partial y & 0 \\ 0 & 0 & \partial N_i / \partial z \\ \partial N_i / \partial y & \partial N_i / \partial x & 0 \\ 0 & \partial N_i / \partial z & \partial N_i / \partial y \\ \partial N_i / \partial z & 0 & \partial N_i / \partial x \end{bmatrix} \quad (i = 1, 2, 3 \dots n) \tag{15}$$

where n denotes the nodal number in an element. Since the shape functions of each element are obtained, the matrix B can be calculated by Eq. (15), and then Eq. (14) can be calculated through Gaussian integral. Finally, the element stiffness matrices for all serendipity elements are obtained.

3.2.3. Assembly of the global stiffness matrix

The core to assemble the global stiffness matrix is to find out the relationship between the local node DOF indices and global node DOF indices. It is equal to construct the relationship between local node identities (IDs) and global node IDs, since DOF indices can be easily derived through node IDs.

Before numbering local node IDs, the type of elements must be known primarily. For fine elements, there is only one type, i.e., 4-node serendipity elements; for middle elements or coarse elements, there are 4-node, 5-node, 6-node, 7-node, and 8-node serendipity elements, and the identification criterion of different types of elements is listed in Table 1. For 5-node, 6-node, and 7-node serendipity elements, they can be subdivided into different types of elements according to the orientation of boundary nodes shown in Fig. 8. In this case, the specific orientation (i.e., down, right, up, and left) of active M-nodes (C-nodes) determines the specific type of 5-node, 6-node, and 7-node serendipity elements. After the types of elements are determined, it is apparent that the element in local node IDs can be easily numbered.

Fig. 13 gives the numbering scheme in three-level meshes with different elemental sizes, which can be stated as follows. Firstly, the node IDs are numbered on the fine mesh one by one. Secondly, when four fine solid or void elements merge into a middle element, remove those node IDs which disappear owing to the merging operation, at the same time, keep other node IDs unchanged. Finally, when four middle elements merge into a coarse element, repeat the previous step. Obviously, this numbering scheme can be extended to 3D cases.

After the local node IDs and global node IDs are numbered, the relationship between the local node IDs and global node IDs can be easily obtained. Taking the coarse element in Fig. 14 as an example, the local node IDs 1, 2, 5, 3, 6 and 4 correspond to the global node IDs 9, 45, 43, 41, 23 and 5. Once the corresponding relationship of each element is known, the global stiffness matrix can be assembled.

As the global node IDs are not numbered in the usual way that their numbers are increased one by one, the DOFs of the disappeared node IDs should be gradually removed in the global stiffness matrix, that is why FEA in the three-level meshes is more efficient.

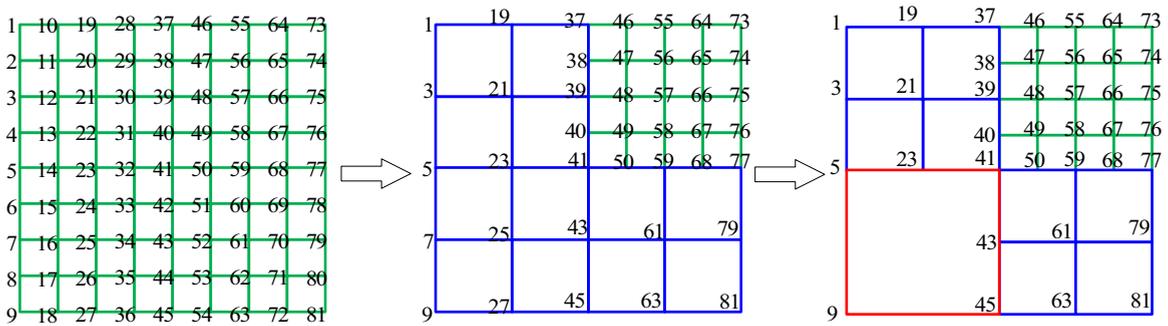


Fig. 13. Numbering the global node IDs in three-level meshes.

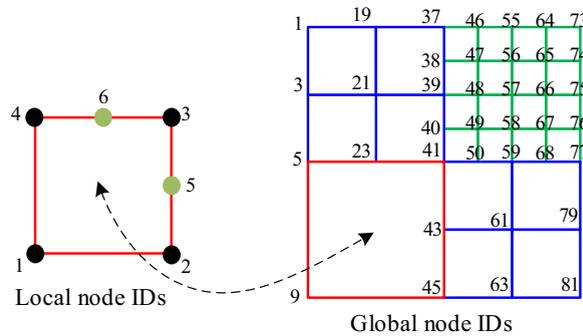


Fig. 14. Relationship between the coarse element local node IDs and global node IDs.

3.3. Sensitivity analysis and filter

To implement the sensitivity filter in the three-level mesh with different element sizes, the three-level mesh is projected to the fine mesh as shown in Fig. 15. A coarse element is projected to 16 identical fine elements and a middle element is projected to 4 identical fine elements (2D problem). Physical quantities to be projected are the element densities and sensitivities. Here, we describe the element densities before and after the projection as x_e and x_E , respectively. For fine elements, the element densities and sensitivities remain unchanged before and after projection. For the middle elements and coarse elements, element densities should also remain unchanged to ensure that the gray levels are consistent during projection. The relationship of element sensitivities during projection can be deduced by the chain rule:

$$\frac{\partial c}{\partial x_E} = \frac{\partial c}{\partial x_e} \frac{\partial x_e}{\partial x_E} = \frac{\partial c}{\partial x_e} \frac{1}{E_{num}} \tag{16}$$

Where E_{num} represents the number of fine elements projected from one element. In 2D cases, E_{num} of fine elements, middle elements and coarse elements are 1, 4 and 16 respectively, while in 3D cases, they are 1, 8, and 64 respectively.

After element densities and sensitivities are projected from the three-level mesh, the sensitivity filter can be performed on the projected fine mesh as that used in conventional topology optimization.

4. Algorithm implementation

This section presents flowchart of the overall algorithm implementation, and several remarks are provided with the purpose of further elaborations about ATMM.

4.1. Flowchart of the whole algorithm

The overall algorithm implementation of ATMM is presented in Fig. 16. The steps with green boxes (i.e., three-level meshing scheme, FEA, and sensitivity analysis and filter) are associated with the proposed ATMM, which will be further discussed. As the other steps are the same as the classic topology optimization, which are not explained in detail here. The detail algorithm implementations of three-level meshing scheme, FEA, and sensitivity analysis and filter are listed in Table 2, Table 3 and Table 4, respectively.

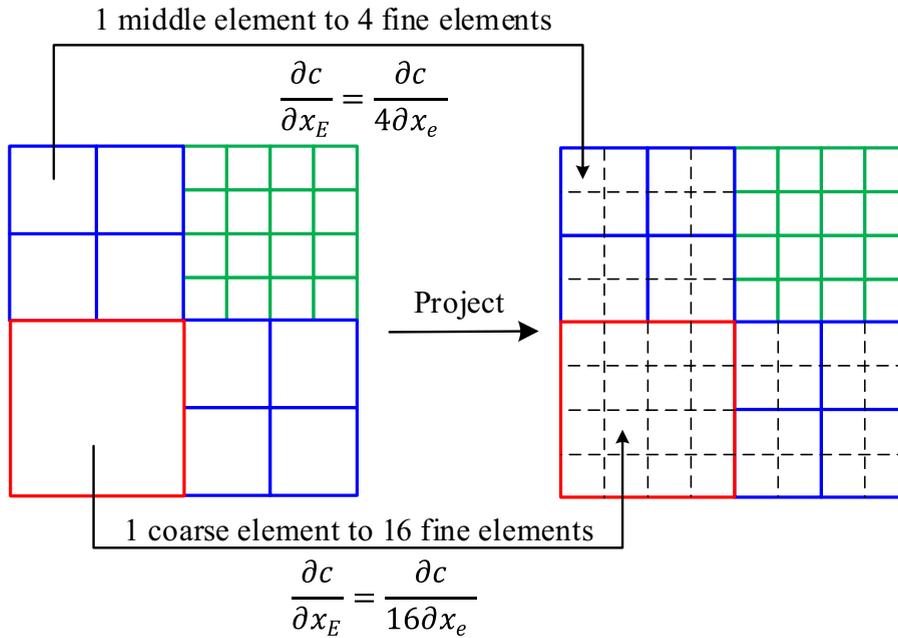


Fig. 15. Three-level mesh is projected to the fine mesh.

Table 2
Algorithm implementation of three-level meshing scheme.

Algorithm 1 Three-level meshing scheme
1: Calculate C-node densities and M-node densities.
2: Find all active M-nodes.
3: Merge the 4 (in 2D) or 8 (in 3D) fine elements around each active M-node into a middle element.
4: Find all active C-nodes.
5: Merge the 4 (in 2D) or 8 (in 3D) middle elements around each active C-node into a coarse element.

Table 3
Algorithm implementation of FEA.

Algorithm 2 FEA
1: Calculate the element stiffness matrices of the 16 (for 2D) and 64 (for 3D) types of serendipity elements according to the shape functions.
2: Determine the type of each element and number the element local node IDs.
3: Number the global node IDs according to the scheme proposed in Section 3.2.3.
4: Calculate the node DOFs according to the node IDs.
5: Assemble the element stiffness matrices into global stiffness matrix according to the relationship between the local node DOFs and global node DOFs.
6: Solve FE equation to obtain the displacement filed.

Table 4
Algorithm implementation of sensitivity analysis and filter.

Algorithm 3 Sensitivity analysis and filter
1: Calculate the sensitivity of each element according to the obtained displacement filed.
2: Project the three-level mesh into the fine mesh, and convert the element densities and sensitivities of the three-level elements into that of fine elements.
3: Perform sensitivity filter on the projected fine mesh as the conventional topology optimization.

4.2. Remarks about ATMM

Four remarks for further elaborations about ATMM are mentioned below.

- (1) ATMM declares that fine elements together with density equal to 0 or 1 can merge into a relatively coarser element. The constraint that the density equals to 0 or 1 can be relaxed with setting upper and lower thresholds. That is, fine elements together with density bigger than the upper threshold or smaller than the lower threshold (e.g., upper threshold equals

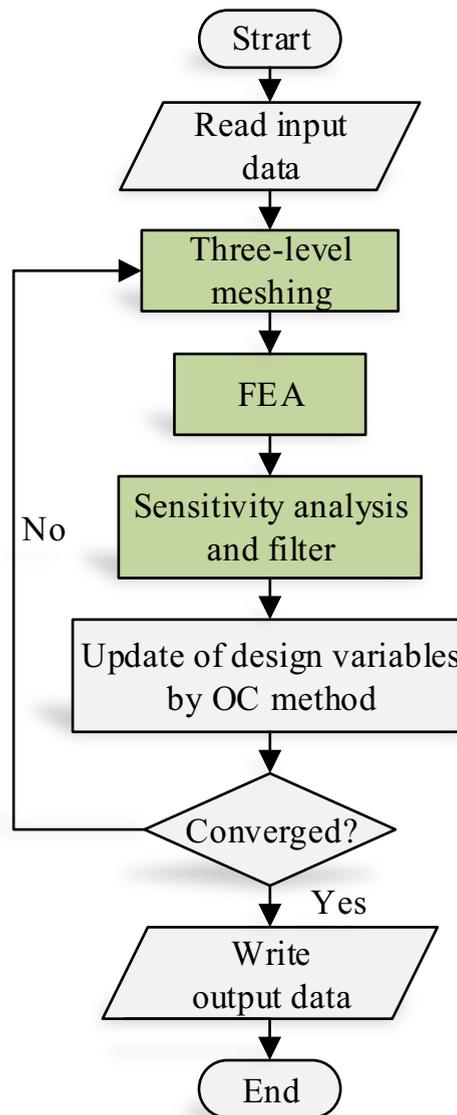


Fig. 16. Flowchart of the overall algorithm implementation.

to 0.99, lower threshold equals to 0.01) can merge into a relatively coarser element. This relaxation would increase the proportion of coarse elements and decrease the DOFs of FEA.

- (2) According to the research in [26], the densities of some coarse elements can be reduced so that the updating of design variables can be avoided, since their element densities keep stable at 0 or 1.
- (3) In addition to the projection method is adopted to implement filter, other filter schemes can be also applied to ATMM, e.g., the filter can works directly on the three-level mesh, through applying large filter radius to the coarse elements and small filter radius to the fine elements, which can be referred to [57].
- (4) ATMM is not limited to the three-level mesh, more levels can be extended, if necessary. Coarse elements can merge into a coarser element and continuously proceed in this way to pursue higher-level element, but it also increases the complexity in programming.

5. Numerical examples

To demonstrate the effectiveness, high efficiency, and accuracy of the proposed ATMM, three benchmark numerical examples including MBB beam, 3D cantilever, L-shaped beam, and compliant mechanism, are tested in this section. In all numerical examples, unless otherwise specified, the Young's modulus of the solid and void material (i.e., E_0 and E_{\min}) are set to 1 and 10^{-9} , respectively, Poisson's ratio is set to 0.3, filter radius is taken as 1.5, volume fraction is set to 0.3. The

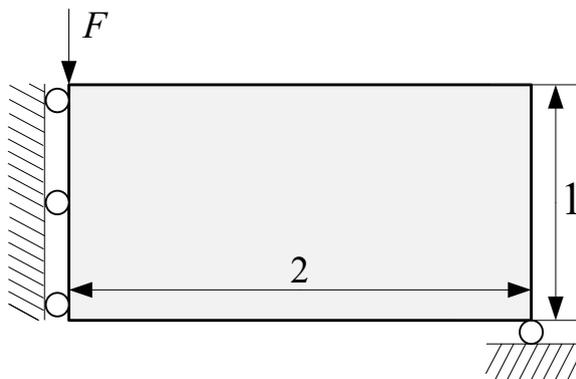


Fig. 17. Design domain, boundary conditions and external force of a half MBB beam.

Table 5
The total time and final compliance comparison between top99 and ATMM.

Total time of top99 (t_1)	Total time of ATMM (t_2)	Speed up t_1/t_2	Compliance of top99 (c_1)	Compliance of ATMM (c_2)	Error $\frac{c_2-c_1}{c_1}$
92.9 second	54.3 second	1.7	127.57	127.65	0.063%

convergence criterion for all examples is: $\max(\text{abs}(x^{(k)} - x^{(k-1)})) \leq 0.01$, i.e., the maximum difference between the design variables in two consecutive iterations is less than 0.01.

The classic top99 code (in 2D case) and its derivative codes (in 3D case) are adopted to compare the numerical performance with the proposed ATMM. Although there are more efficient top88 code [58] and its 3D form [53], their high efficiency is just implemented by the efficient use of MATLAB, i.e., loop vectorization, instead of improving the efficiency in the algorithm themselves. The three-level mesh algorithm speeds up topology optimization fundamentally, instead of the vectorization in MATLAB. Therefore, the proposed ATMM is a general algorithm. When efficient programming languages (e.g., C++ and Fortran) are used, it is believed that the proposed algorithm can perform better than top88 in terms of computational efficiency due to the reduction of DOFs. In this paper, all the codes are run on MATLAB, and thereby it is fair to compare the proposed ATMM with top99 and its derivative 3D codes without loop vectorized.

5.1. MBB beam

MBB beam is utilized to verify the accuracy and efficiency of the proposed method. The design domain, boundary conditions and external force of a half MBB beam are illustrated in Fig. 17. The design domain is discretized into 120×60 elements, which is the initial fine mesh with uniform elements in level 1.

Fig. 18 (a) depicts the optimized structures with some intermediate optimization steps and the converged step for top99, and Fig. 18 (b) shows the optimized structures and meshes with the same intermediate steps and the converged step for ATMM. As shown in Fig. 18, the material distribution optimized by ATMM is consistent with the standard top99 code in different iteration steps, which demonstrates the effectiveness of the proposed method. Besides, there is little difference in convergence performance between ATMM and top99, top99 obtains the final structure at step 288, while ATMM is at step 286. Additionally, we can find that the structural mesh in the first step is composed of fine elements (in Fig. 18 (b)), as optimization progresses, the structures gradually become white and black and the mesh is composed of higher-level elements (i.e., middle elements and coarse elements). The fine elements in ATMM are mainly distributed at the structural boundaries, i.e., transition regions between black and white areas, which proves the promise of the meshing scheme, as boundary areas do need to be described by fine elements for obtaining accurate structures in density-based topology optimization. Since the higher-level elements are applied in the solid or void area of a structure instead of uniform fine elements used in top99, it can be deduced that the structural DOFs in ATMM are fewer than top99, and therefore the time cost in each iteration is less than top99.

The quantitative comparisons in DOF and time between top99 and ATMM are depicted in Fig. 19. In Fig. 19, DOF percentage means the percentage of ATMM DOFs with respect to that of top99 during iterations, as the optimization progresses, the DOF percentage gradually decreases from initial 100% to about 50% at the end due to the three-level meshing scheme. It is also observed that the 50% DOFs retains a relative long iteration period approximately from step 125 to the converged step 286, which guarantees the high efficiency of ATMM. Time comparison shows that the computational time of ATMM in each iteration is less than top99 due to the DOF reduction, with the 50% DOFs reduction, the time in each iteration can be reduced up to 47% compared to top99, which proves the high efficiency of ATMM.

To further demonstrate the efficiency and accuracy of the proposed method, the total running time during optimization and the final compliance of top99 and ATMM are listed in Table 5. The speed up value of total time is 1.7 with only

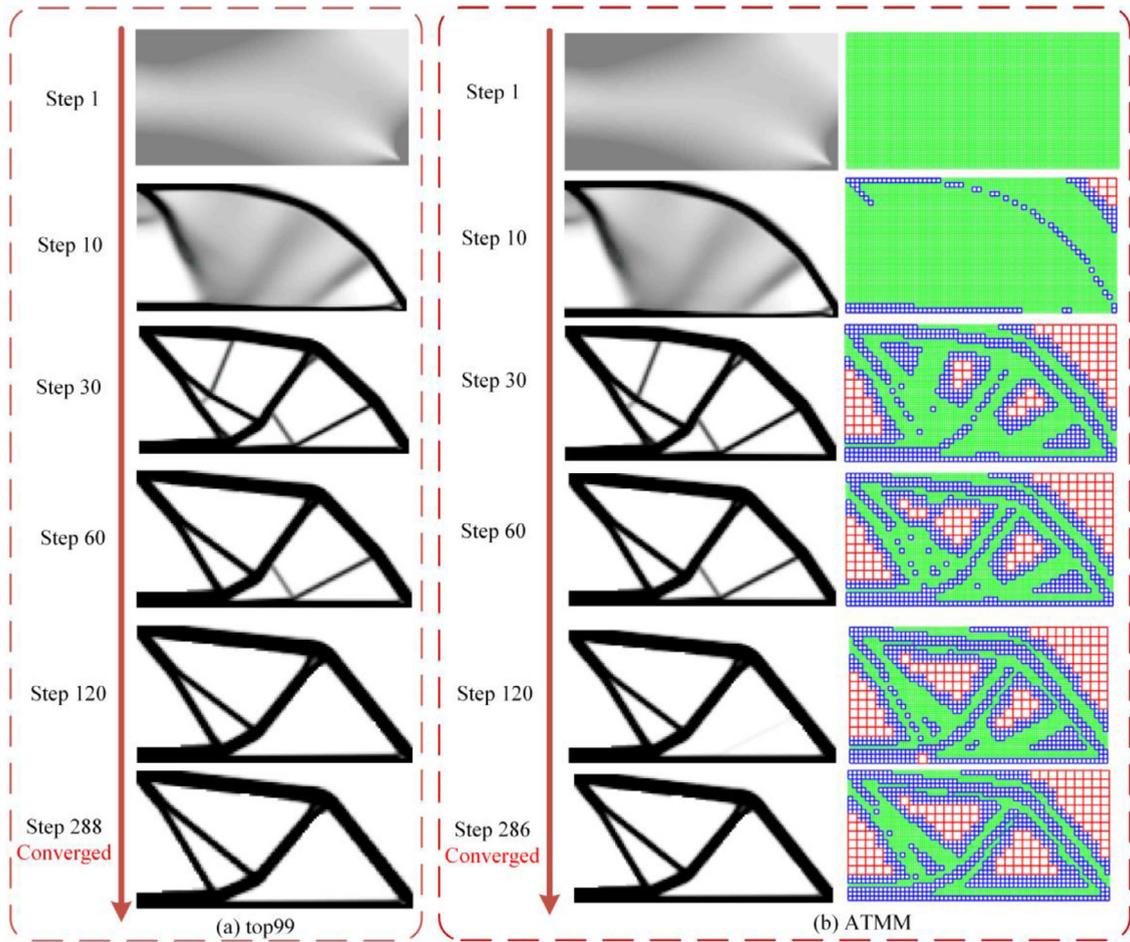


Fig. 18. The optimized MBB structures at different steps: (a) top99, and (b) ATMM (left) with the applied mesh (right).

0.063% objective function (compliance) error, which demonstrates that the three-level algorithm not only possesses of high computational efficiency, but also has a reliable optimization result.

In SIMP-based topology optimization, it has been proved that there are still some intermediate densities in the final structure, even though the penal factor is introduced [59]. In Fig. 19, it is found that about 50% DOFs are reduced in the mid-late iterations, which is in the case that quite a few intermediate densities exist in the corresponding structures. It is believed that DOFs will be further reduced if a gray-scale suppression method is introduced, since more coarse and middle elements can be formed. Here, the gray-scale suppression method implemented by modified OC is adopted [26], such as

$$x'_{new} = \frac{1}{1 + e^{-\alpha(x_{new} - 0.5)}} \tag{17}$$

where x_{new} is the design variables updated by OC, x'_{new} is the final design variables. It means that the final design variables are firstly updated by OC and then obtained through Eq. (17), which is called the modified OC. The coefficient α is a parameter determining the degree of gray-scale. In this paper, we set α as a constant value of 15. As declared in this method, it should be applied in the late stage of topology optimization since the topological structure is not stable in the early stage. The late stage of topology optimization in this paper can be identified by the relative variation of DOF percentage, which can be expressed as

$$DOF_{ch} = \frac{\left| \sum_{i=1}^3 (DOF_{per}^{(k-i+1)} - DOF_{per}^{(k-2-i)}) \right|}{\left| \sum_{i=1}^3 DOF_{per}^{(k-i+1)} \right|} \tag{18}$$

where DOF_{ch} represents the relative variation of DOF percentage; k denotes the current iteration number; DOF_{per} is the DOF percentage of ATMM with respect to top99. When DOF_{ch} is smaller than a small threshold ξ (e.g., ξ is set to

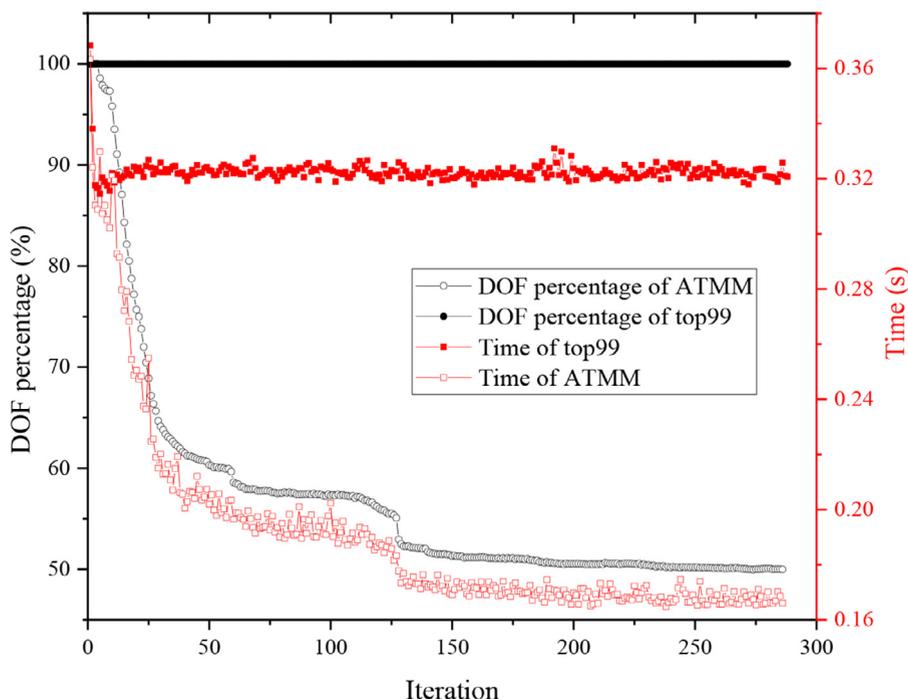


Fig. 19. Time and DOF comparisons between top99 and three-level meshing algorithm.

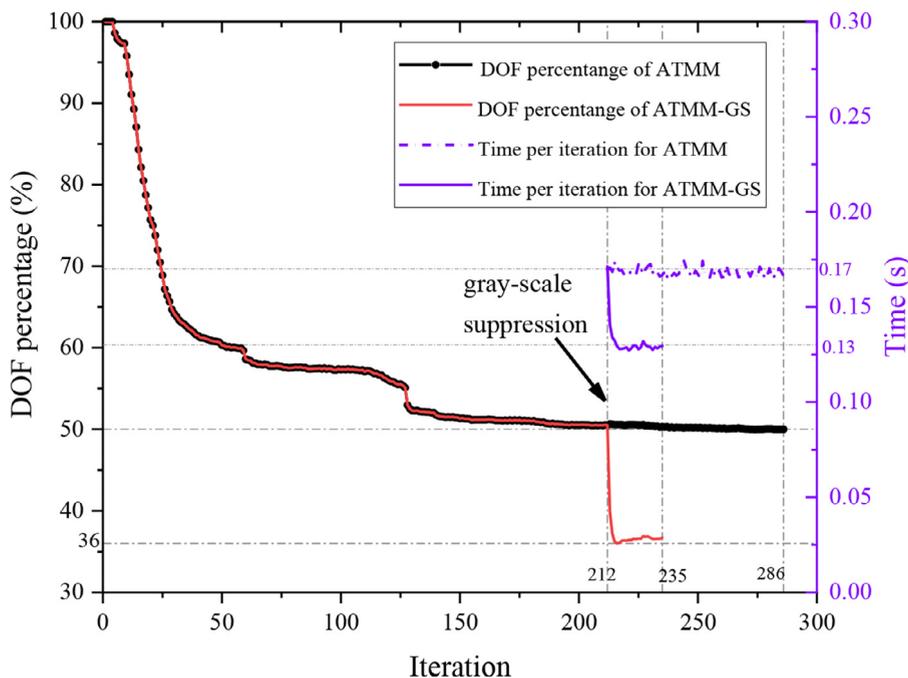


Fig. 20. Comparisons of DOF percentage and time per iteration between ATMM and ATMM-GS.

0.00005), i.e., the DOFs of the mesh are almost unchanged, the gray-scale suppression starts to be employed to suppress the gray elements, such that the DOFs can be further reduced.

Hereafter, for the convenience of description, the ATMM with gray-scale suppression is terms as ATMM-GS. Fig. 20 shows the DOF percentage and time per iteration comparisons between ATMM and ATMM-GS during optimization. It can be seen that the DOFs can be further reduced from 50% to 36% after the gray-scale suppression is employed at iteration 212, and the time in each iteration starts to be reduced from 0.17s to 0.13s. Additionally, we can also find that the convergence with

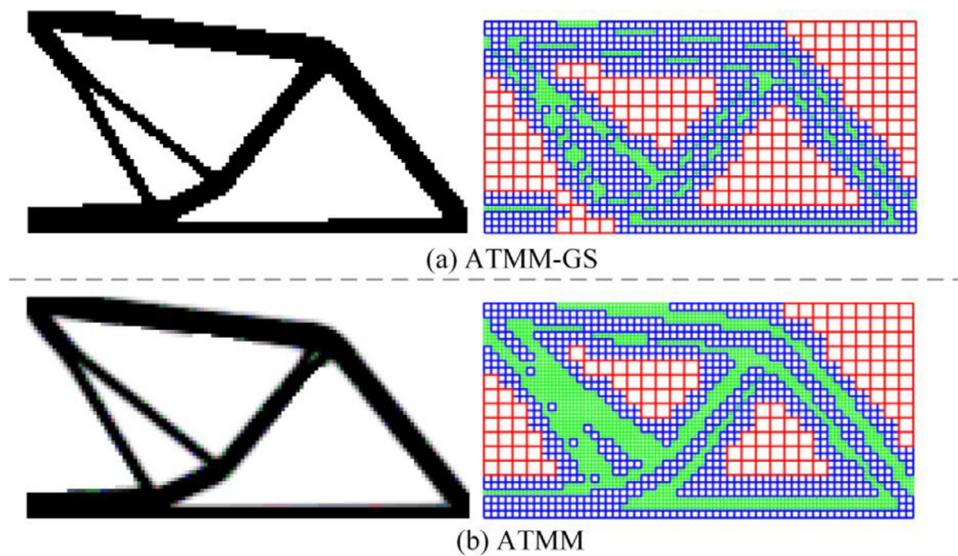


Fig. 21. Comparisons of the final optimized structure and its employed mesh between ATMM and ATMM-GS.

Table 6

Total time, final compliance, iteration number, and achieved minimum DOF percentage of top99, ATMM, and ATMM with gray-scale suppression.

Algorithms	Compliance	Iteration number	Achieved minimum DOF percentage	Time per iteration for the minimum DOFs (s)	Total time (s)
Top99	127.57	288	100%	0.32	92.9
ATMM	127.65	286	50%	0.17	54.3
ATMM-GS	120.40	235	36%	0.13	44.8

gray-scale suppression (converged at iteration 235) performs better than that without gray-scale suppression (converged at iteration 286). This phenomenon has the generality as more intermediate elements are suppressed, and the densities of most elements keep stable around 0 or 1, which is especially beneficial to meet the convergence criterion: $\max(\text{abs}(x^{(k)} - x^{(k-1)})) \leq 0.01$.

Comparisons of the final optimized structure and its employed mesh between ATMM and ATMM-GS are shown in Fig. 21. It can be seen that the final topological solution of ATMM-GS is almost the same as that of the classic top99 and AMTA, which demonstrates that the introduction of gray-scale suppression has negligible influence on the topological structure. Owing to the gray-scale suppression, the final structure becomes nearly black and white without gray elements, and therefore, there are more high-level elements in the mesh of structure, which means fewer DOFs are obtained. It arrives at the preliminary conclusion that the gray-scale suppression can cooperate well with ATMM in terms of computational efficiency and optimization accuracy.

For comprehensive comparisons,

Table 6 presents the final compliance, iteration number, achieved minimum DOF percentage, time in each iteration when the minimum DOF percentage is achieved, and the total time of the three different optimization algorithms (i.e., top99, ATMM, ATMM-GS). As shown in

Table 6, the following four conclusions can be drawn:

- (1) Due to three-level meshing scheme, the ATMM can reduce the DOFs of a structure up to 50%. Moreover, when combined with the gray-scale suppression, the DOFs can be further reduced from 50% to 36% since more gray elements are suppressed, and then more elements become solids or voids, so that the mesh of a structure is composed of coarser elements with fewer DOFs.
- (2) ATMM-GS has better performance than top99 in terms of convergence speed, this can be explained as the gray-scale suppression further pushes the design variables towards 1/0, and therefore, the convergence condition can be easier to satisfy.
- (3) The final compliance obtained by ATAM has negligible error compared to the standard top99. Furthermore, when working with the gray-scale suppression, as more gray elements are suppressed the structural materials are more fully used so that the compliance becomes less than top99, which means the better compliance is acquired.
- (4) Due to the three-level meshing scheme, the time in each iteration is just 53% (i.e., 0.17/0.32) that of top99, with the help of the gray-scale suppression, the time in each iteration can be further reduced to 41% (i.e., 0.13/0.17) that of top99. In terms of the total time, the ATMM has the speedup ratio of 1.7 compared to top99 as the time in each iteration is

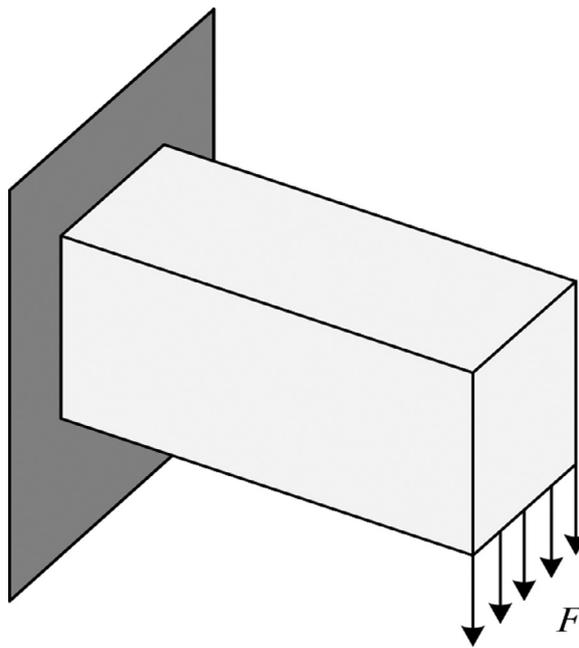


Fig. 22. Design domain, boundary constraint, and applied load for 3D cantilever.

reduced; the ATMM-GS has the speedup ratio of 2.1 compared to top99, which is not only due to the reduced time in each iteration but also owing to the reduced number of iterations.

In virtue of the excellent performance of ATMM-GS in MBB beam, ATMM is recommended to combine with gray-scale suppression for a better effect. In the following numerical examples, the numerical performance of ATMM-GS will be further demonstrated.

5.2. 3D cantilever

This section selects the 3D cantilever to examine the performance of the ATMM-GS in 3D topology optimization. The design domain, boundary condition and the applied load is shown in Fig. 22. The initial mesh is with a size of $60 \times 40 \times 4$ to discretize the design domain.

Fig. 23 presents some intermediate optimized structures and the final optimized structure for top3d and ATMM-GS, as well as the mesh applied to the structures at different steps in ATMM-GS. The usual image display in top3d code only displays part of the structure with densities greater than 0.5 [53], the structure in Fig. 23 is completely displayed whatever the densities are. So, it is reasonable to see that the structures in Fig. 23 have quite a few low-density elements. In Fig. 23 (b), before step 57, the ATMM-GS algorithm is performed by ATMM only, when the relative variation of DOFs becomes smaller than the threshold (i.e., $DOF_ch < \xi$), gray-scale suppression is introduced, ATMM-GS works at step 58 until convergence.

As shown in Fig. 23, the intermediate structures before step 57 optimized by top3d and ATMM-GS are consistent, which verifies the effectiveness of the ATMM in 3D topology optimization. After step 58, the gray-scale suppression is introduced, it can be clearly seen that the low-density elements are vanished and the structures become black and white. In top3d, there are many low-density elements emerged in the intermediate structures and the final optimized structure, so it is of necessity to introduce gray-scale suppression to overcome this defect. Comparing the final structure between top3d and ATMM-GS, the topological structure in ATMM-GS is the same as the black part in top3d, which dispels the worries that the introduction of three-level mesh and gray-scale suppression may lead to the wrong optimization result. Besides, ATMM-GS greatly accelerates the convergence speed based on the fact that top3d satisfies the convergence condition at step 288, while ATMM-GS converges at step 74, which can be explained as the polarization effect on densities. Furthermore, seen from the adaptive three-level mesh shown in Fig. 23 (b), the DOF percentage and time in each iteration will gradually decrease, whose curves are depicted in Fig. 24. In Fig. 24, the DOF percentage in ATMM-GS can be reduced from 100% to 60% before gray-scale suppression, and it can be further reduced to 38% after gray-scale suppression used. The corresponding time in each iteration can be reduced from about 1.35s in top3d to 0.75s before gray-scale suppression, and finally reduced to 0.47s after gray-scale suppression. In the 2D case (120×60) illustrated in Section 5.1, the maximum saving time in each iteration is 0.19s, while in the 3D case ($60 \times 40 \times 4$) it is 0.88s, implying the time consumed in each iteration can be saved more in large-scale problems.

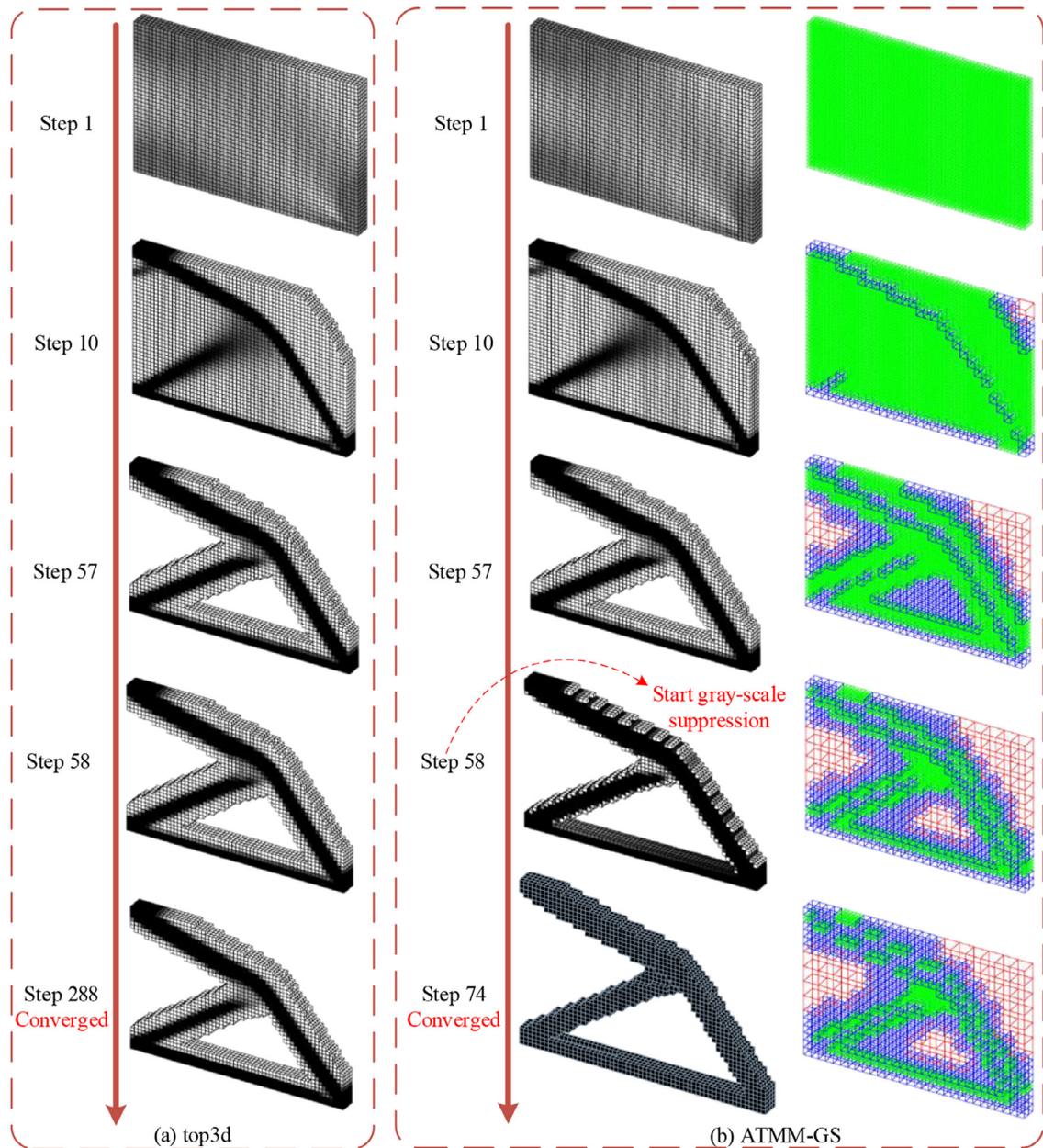


Fig. 23. The optimized 3D cantilever structures at different steps: (a) top3d, and (b) ATMM-GS (left) with the applied mesh (right).

Comparisons of the final compliance, total time, iteration number as well as the average time per iteration between top3d and ATMM-GS are presented in Table 7. Besides, for a fully comparison, Table 7 also presents the results of other two mesh scales: $100 \times 60 \times 4$ and $100 \times 60 \times 8$.

As shown in Table 7, the compliances of ATMM-GS are all less than that of top3d, which reflects that ATMM-GS has the ability to find the better material distribution compared to top3d because of the gray-scale suppression. The iteration number of ATMM-GS is also greatly reduced, which is about half of top3d. The average time in each iteration (i.e., the value of total time divided by iteration number) shows that ATMM-GS can save about 37%-71% compared to the time of top3d, which shows the admirable performance of ATMM-GS in terms of computational efficiency. Moreover, as the mesh scale increases, the average time saved in each iteration becomes greater, which illustrates the ability of ATMM-GS to handle large-scale problems. Due to the reduced iteration number and average time in each iteration, the total time of ATMM-GS is much reduced, and the speedup ratios range from 4.2 to 6.0 under three different mesh scales, demonstrating the high efficiency of ATMM-GS.

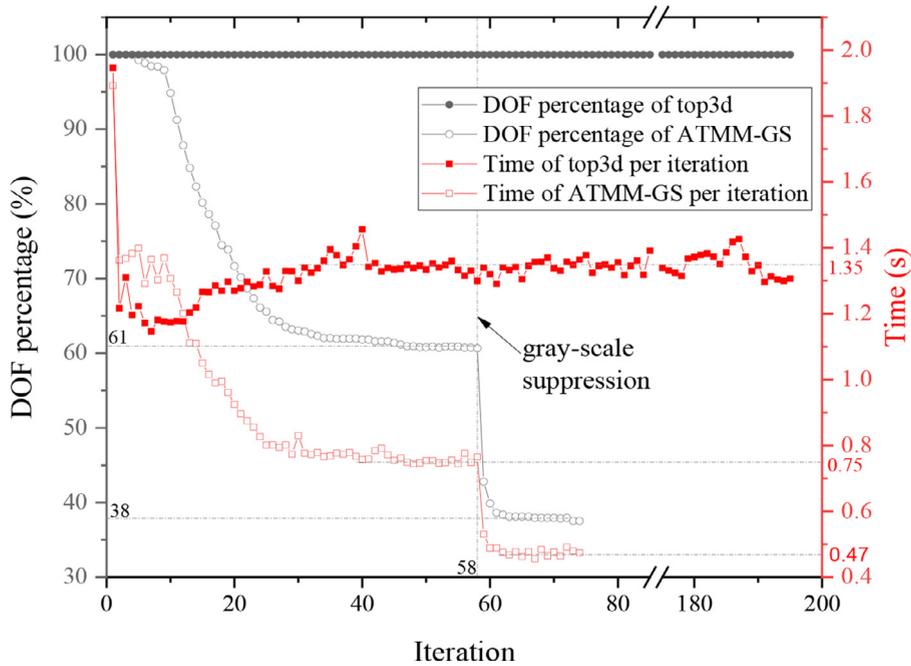


Fig. 24. Curves of DOF percentage and time with iteration of top3d and ATMM-GS.

Table 7

Comparisons of the final compliance, total time, iteration number as well as the average time in each iteration between top3d and ATMM-GS.

Mesh scale	Compliance ATMM-GS (top3d)	Iteration number ATMM-GS (top3d)	Total time (s) ATMM-GS (top3d)	Average time in each iteration (s) ATMM-GS (top3d)
60 × 40 × 4	380.50 (410.75)	74(195)	62.0 (260.2)	0.838 (1.334)
100 × 60 × 4	468.55 (498.79)	145 (289)	263.9 (1241.8)	1.820 (4.297)
100 × 60 × 8	735.36 (790.04)	129 (224)	631.8 (3762.6)	4.898 (16.797)

Table 8

The comparisons of efficiency and compliance between ATMM-GS and top3d.

Mesh scale	Compliance ATMM-GS (top3d)	Iteration number ATMM-GS (top3d)	Total time (s) ATMM-GS (top3d)	Average time in each iteration (s) ATMM-GS (top3d)
80 × 80 × 20	3594.31(3921.06)	204 (354)	2748.53 (15391.69)	13.47 (43.48)

In conclusion, this example has proved two advantages of ATMM-GS compared to top3d: one is the computational efficiency is much improved reflected by the reduction of iteration number, time in each iteration and total time; the other is the reliable optimization results reflected by the consistent topological configurations and better compliances.

5.3. L-shaped beam

In this section, the L-shaped beam is investigated to verify the efficiency of ATMM-GS. The design domain, external load, and boundary constraint are shown in Fig. 25. The design domain is initially discretized with 80 × 80 × 20 elements, where 40 × 40 × 20 elements are suppressed. The volume fraction is set to 0.15, the Young’s moduli and Poisson ratio remain unchanged.

Fig. 26 presents the final optimized topology structures of ATMM-GS and top3d. It can be seen that the optimized structure of ATMM-GS is almost the same as that of top3d, which demonstrates the effectiveness of ATMM-GS in the L-shape beam. The curves of DOF percentage and time with iteration for top3d and ATMM-GS are depicted in Fig. 27. It can be found that as the DOF percentage in ATMM-GS gradually decreases, the time cost in each iteration decreases as well. The comparisons of efficiency and compliance between ATMM-GS and top3d are listed in Table 8. Compared to top3d, ATMM-GS achieves a speedup of 3.2 times on the average time per iteration due to the application of three-level mesh, and has the speedup of 5.6 on the total time since both the DOF percentage per iteration and iteration number are reduced. ATMM-GS has obvious advantages over the top3d, in terms of the average time per iteration and the total time. Besides, ATMM-GS has

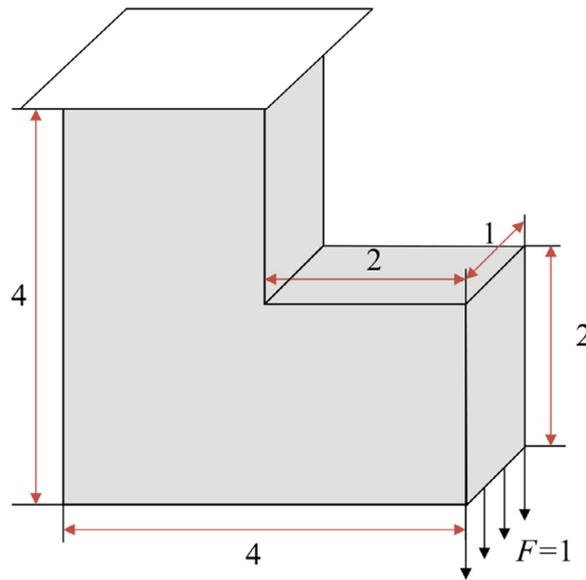


Fig. 25. The design domain, external load, and boundary constraint of L-shaped beam.

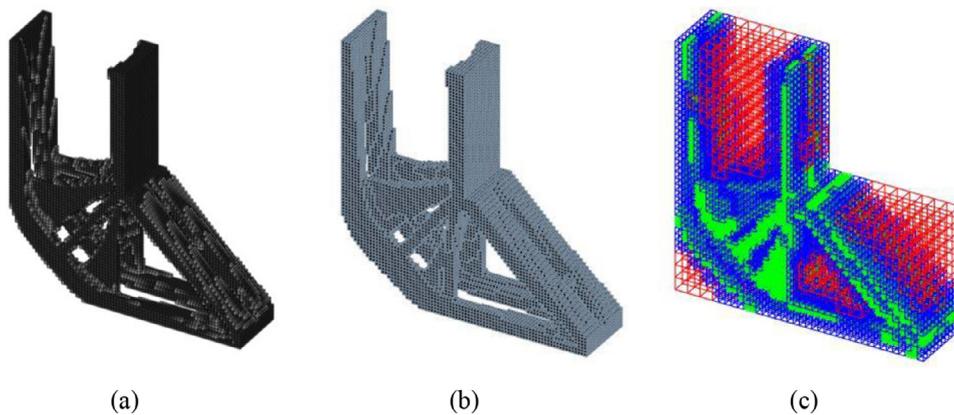


Fig. 26. The final optimized structures of top3d and ATMM-GS. (a) the structure optimized by top3d. (b) the structure optimized by ATMM-GS. (c) the FE mesh applied in the ATMM-GS

less compliance than top3d, which further demonstrates the accuracy in ATMM-GS. So ATMM-GS can significantly improve the topology optimization efficiency with reliable topologies and accurate objective compliances.

5.4. Compliant mechanism

A 3D compliant mechanism is employed to illustrate the efficiency of ATMM-GS in obtaining the design of force inverter. Fig. 28 presents the design domain, boundary conditions, and external loads.

Different from the minimum compliance problem, the objection function of the compliant mechanism is to maximize the negative horizontal output displacement induced by a positive input external force. The mathematical model can be written as follows:

$$\begin{aligned}
 &\text{Find : } x = (x_1, x_2, x_3, \dots, x_N)^T \\
 &\text{Min : } c(x) = -u_{out} = -L^T U = U_d^T K U \\
 &\text{s.t. : } \begin{cases} V(x) \leq V_{of} \\ K U = F \\ 0 \leq x \leq 1 \end{cases} \tag{19}
 \end{aligned}$$

Where u_{out} is the output displacement, L is a vector with 0 at all DOFs except 1 at the DOF of the output point, U_d is a global adjoint vector which is the solution of Eq. (20). The other variables are the same as Eq. (2). The sensitivity of objective

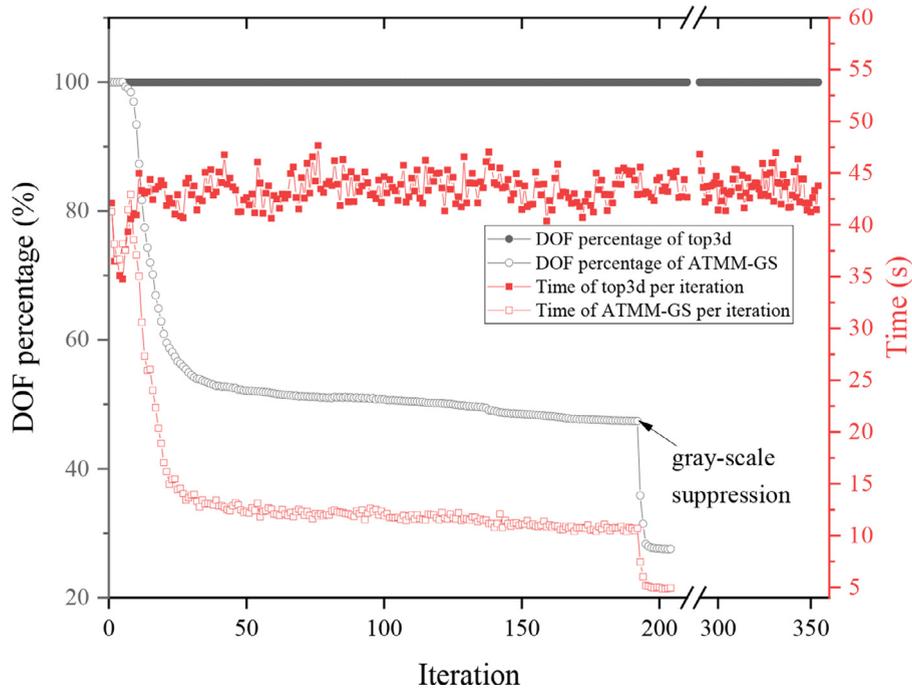


Fig. 27. The curves of DOF percentage and time with iteration of top3d and ATMM-GS

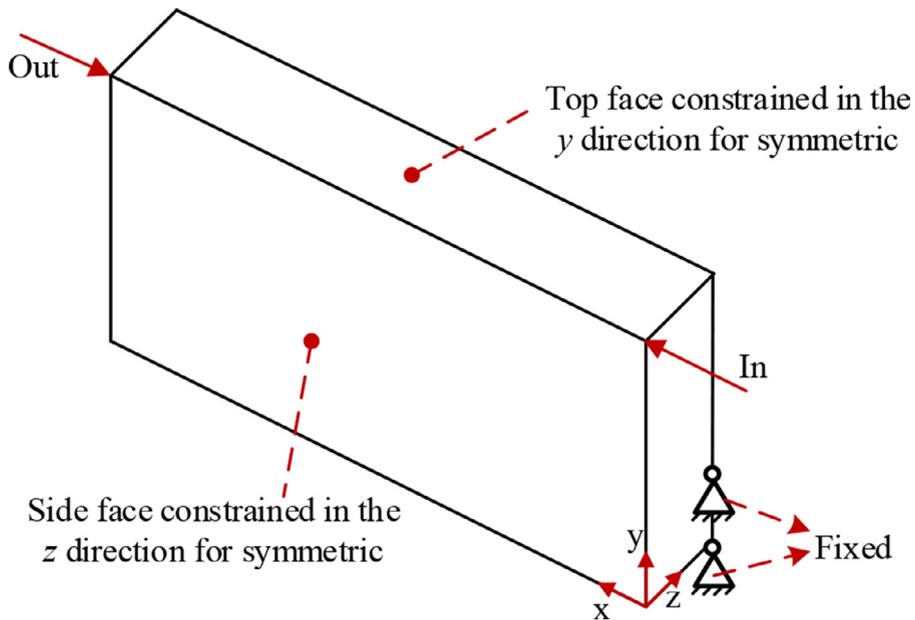


Fig. 28. Illustration of a 3D force inverter.

function with respect to element density can be deduced as Eq. (21), where u_{de} is the part of U_d associated with element e

$$KU_d = -L \tag{20}$$

$$\frac{\partial c}{\partial x_e} = -px_e^{p-1} (E_0 - E_{\min}) u_{de}^T k_e u_e \tag{21}$$

The code derived from top3d is utilized as the comparison of ATMM-GS, which is named as topCM. Considering that the compliant mechanism problem is hard to converge, the maximum iteration number is set to 300. The design domain is initially discretized into $60 \times 40 \times 8$ elements.

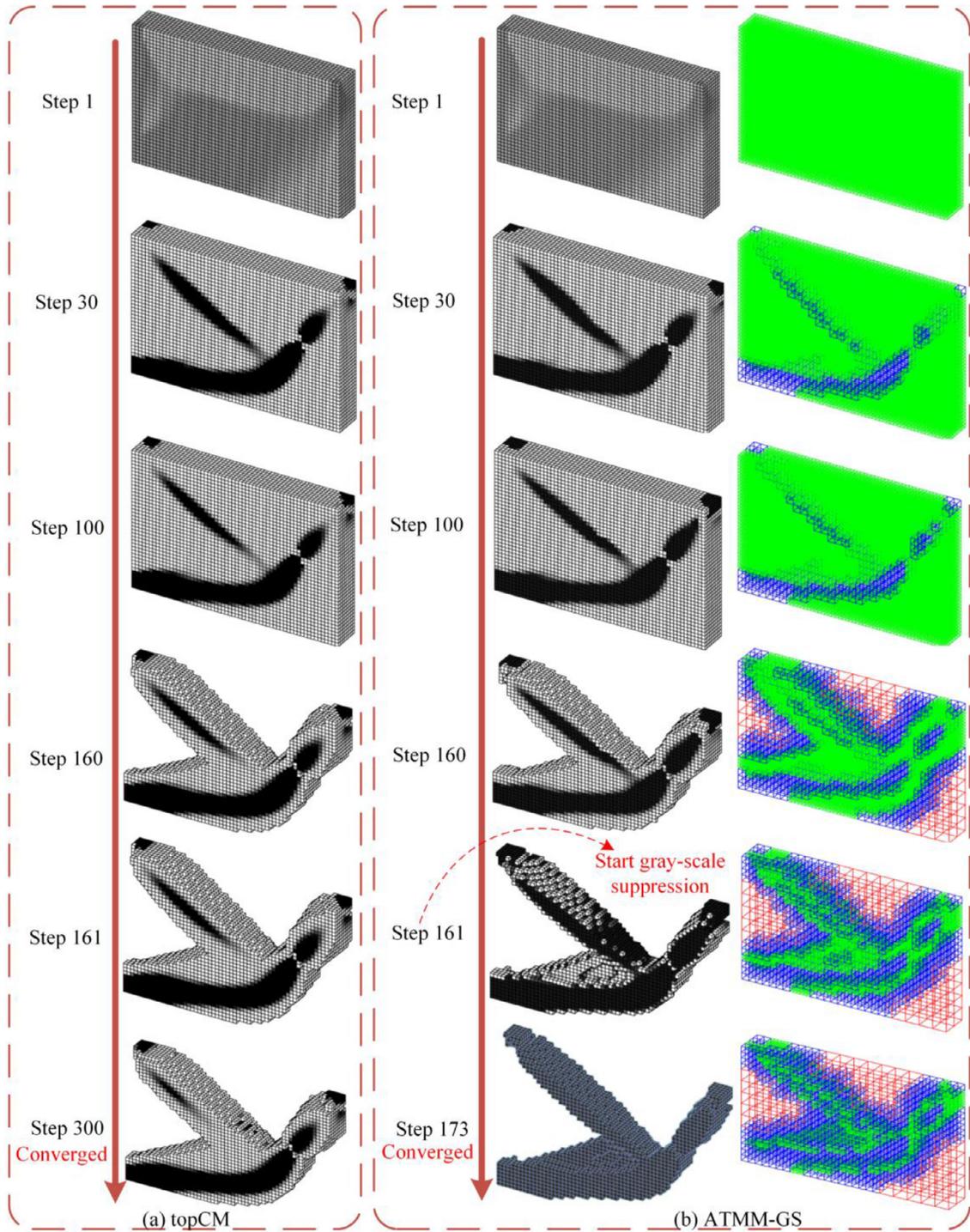


Fig. 29. The optimized 3D compliant mechanism structures at different steps: (a) topCM, and (b) ATMM-GS (left) with the applied mesh (right).

Fig. 29 presents some optimized structures for topCM and ATMM-GS. It can be seen that the ATMM-GS can be well applied to the compliance mechanism problem. The mesh adapts well to topological structure at each step, i.e., the solid or void areas are exactly filled with relatively coarse elements, which prove the good adaptability of three-level mesh. Before step 160, the structures optimized by ATMM-GS are nearly the same as those obtained by topCM. After step 160, the gray-scale suppression is introduced to make the intermediate elements polarization to avoid a few visible low-density elements exist in the final structure of topCM shown in Fig. 29 (a). Besides, the final structures of topCM and ATMM-GS are almost the

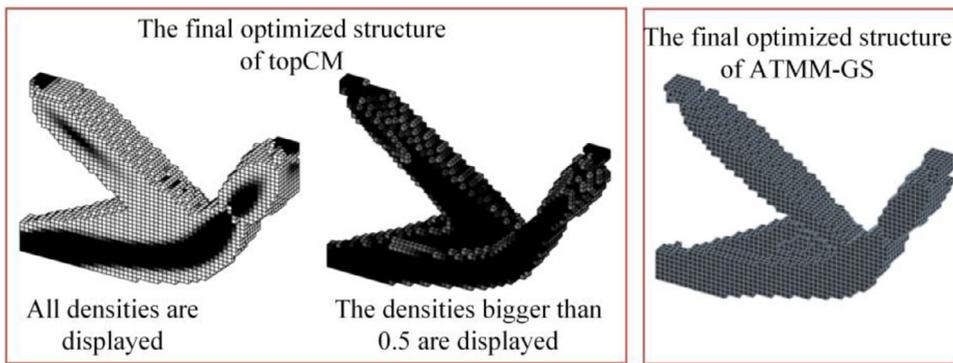


Fig. 30. Comparison of the final optimized structure between topCM and ATMM-GS.

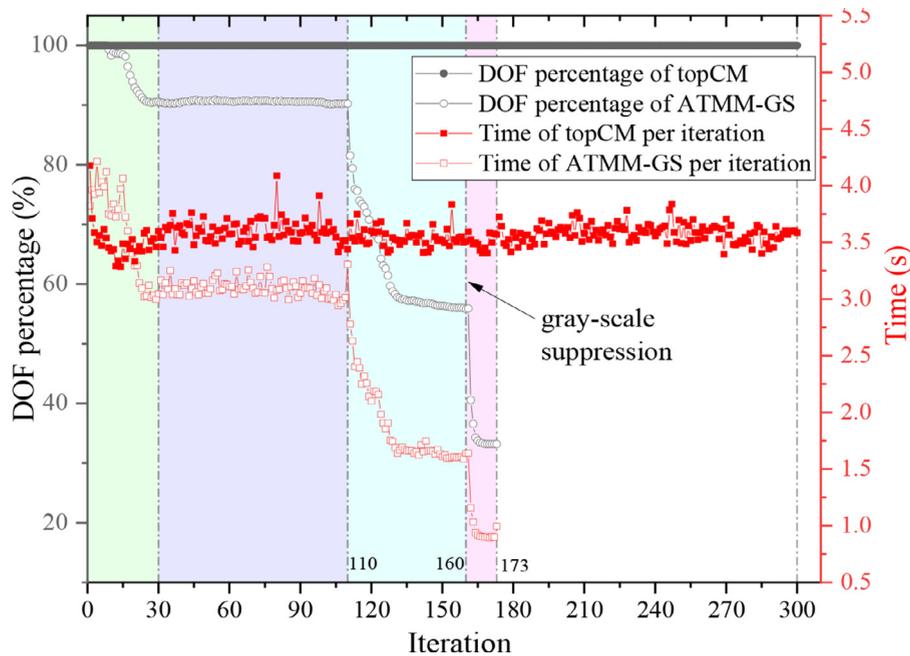


Fig. 31. Curves of DOF percentage and time in each iteration for topCM and ATMM-GS.

Table 9
Comparisons of time and displacement between topCM and ATMM-GS.

Algorithm	Displacement	Iteration number	Total time (s)	Average time (s)
TopCM	-1.7667	300	1428.1	4.76
ATMM-GS	-1.8805	173	460.4	2.66

same, as shown in Fig. 30, which demonstrates the effectiveness of ATMM-GS in obtaining favorable topological structures of compliance mechanism.

Fig. 31 shows the curves related to DOF percentage and time in each iteration for topCM and ATMM-GS. At the initial stage, the outline of the optimized structure gradually appeared (see step 1~30 in Fig. 29 (a)), black area was filled with relatively coarse elements so that the DOFs gradually decrease to 90%. Next, the topological structure has nearly no change (see step 31~100 in Fig. 29 (a)) so that the DOFs remain stable during these iterations. In step 110~160, gray elements in topological structures gradually vanish (see step 100~160 in Fig. 29 (a)) so that DOFs are decreased to 30%. Lastly, the gray-scale suppression is introduced, gray elements are further reduced (see step 160~173 in Fig. 29 (a)) so that the DOFs are further reduced to only about 20%. Due to the DOFs reduction, it can be seen that the time in each iteration of ATMM-GS is less than that of topCM, which demonstrates the high efficiency of ATMM-GS. To quantitatively describe this, the total time of the whole optimization process and average time in each iteration are tested, these results are listed in Table 9. To demonstrate the accuracy of optimization result of ATMM-GS, Table 9 also presents the output displacements calculated by

ATMM-GS and topCM. The total time of ATMM-GS only accounts for 32.2% that of topCM, and the average time consumed in each iteration only accounts for 55.9% that of topCM. Moreover, ATMM-GS can obtain better output displacement. It can be concluded that the ATMM-GS outperforms than traditional topCM in terms of computational efficiency and accuracy.

6. Conclusions

This paper proposed an adaptive three-level mesh method (ATMM) without loss of accuracy, to accelerate the structural topology optimization. The key issue of ATMM is to coarsen the elements those are fully voids or solids, and simultaneously remain the fine elements in the transition regions between solids and voids. The three-level meshing scheme is implemented by the mutual transformations between low-level elements and high-level elements. The 16 (2D case) and 64 (3D case) types of serendipity elements are introduced to preserve the elemental compatibility between different levels, and a nodal numbering scheme is given to assemble the global stiffness matrix. The elemental densities and sensitivities in the three-level meshes are projected to the uniform fine meshes, such that the sensitivity filter and OC updating are performed as the conventional topology optimization. Additionally, ATMM is recommended to combine with the gray-scale suppression technology to further reduce the computational costs. Four benchmark examples have been provided to demonstrate that ATMM can greatly improve the computational efficiency, both the topological layouts and objective function values are accurately reserved, which shows that ATMM has bright prospects to solve the large-scale problems. The variable-node elements [60,61], which can have arbitrary nodes on the element side, are taken into consideration to implement a more flexible mesh refinement for more complex topology layout in our future work. The combination of parallel computing and ATMM, and the unstructured meshes or meshes with other element types are also taken into considerations.

Replication of results

All the data used to support the findings of this study are available from the corresponding author upon request.

Declarations Competing of Interest

The authors declare that they have no conflict of interest.

Acknowledgments

This work has been supported by The National Key R&D Program of China (2020YFB1708300), [National Natural Science Foundation of China \(52075184\)](#), and Guangdong Basic and Applied Basic Research Foundation (2019A1515011783, 2019A1515110183). These supports are gratefully acknowledged.

References

- [1] M.P. Bendsøe, N. Kikuchi, Generating optimal topologies in structural design using a homogenization method, *Comput. Method Appl. Mech. Eng.* 71 (1988) 197–224.
- [2] S. Zhang, H. Li, Y. Huang, An improved multi-objective topology optimization model based on SIMP method for continuum structures including self-weight, *Struct. Multidiscip. Optim.* (2020) 1–20.
- [3] W. Zuo, K. Saitou, Multi-material topology optimization using ordered SIMP interpolation, *Struct. Multidiscip. Optim.* 55 (2017) 477–491.
- [4] Y. Zheng, Y. Wang, X. Lu, Z. Liao, J. Qu, Evolutionary topology optimization for mechanical metamaterials with auxetic property, *Int. J. Mech. Sci.* 179 (2020) 105638.
- [5] Y. Wang, D.J. Benson, Geometrically constrained isogeometric parameterized level-set based topology optimization via trimmed elements, *Front. Mech. Eng.* 11 (2016) 328–343.
- [6] X. Xie, S. Wang, M. Xu, Y. Wang, A new isogeometric topology optimization using moving morphable components based on R-functions and collocation schemes, *Comput. Methods Appl. Mech. Eng.* 339 (2018) 61–90.
- [7] Y. Luo, J. Xing, Z. Kang, Topology optimization using material-field series expansion and Kriging-based algorithm: An effective non-gradient method, *Comput. Methods Appl. Mech. Eng.* 364 (2020) 112966.
- [8] X. Zhang, J. Xing, P. Liu, Y. Luo, Z. Kang, Realization of full and directional band gap design by non-gradient topology optimization in acoustic metamaterials, *Extrem. Mech. Lett.* 42 (2021) 101126.
- [9] A. Pizzolato, A. Sharma, R. Ge, K. Maute, V. Verda, A. Sciacovelli, Maximization of performance in multi-tube latent heat storage – optimization of fins topology, effect of materials selection and flow arrangements, *Energy* 203 (2020) 114797.
- [10] S. Nishi, T. Yamada, K. Izui, S. Nishiwaki, K. Terada, Isogeometric topology optimization of anisotropic metamaterials for controlling high-frequency electromagnetic wave, *Int. J. Numer. Methods Eng.* 121 (2020) 1218–1247.
- [11] G. Fujii, M. Takahashi, Y. Akimoto, CMA-ES-based structural topology optimization using a level set boundary expression—Application to optical and carpet cloaks, *Comput. Methods Appl. Mech. Eng.* 332 (2018) 624–643.
- [12] L. Chen, C. Lu, H. Lian, Z. Liu, W. Zhao, S. Li, H. Chen, S.P.A. Bordas, Acoustic topology optimization of sound absorbing materials directly from subdivision surfaces with isogeometric boundary element methods, *Comput. Methods Appl. Mech. Eng.* 362 (2020) 112806.
- [13] G.A. da Silva, A.T. Beck, O. Sigmund, Stress-constrained topology optimization considering uniform manufacturing uncertainties, *Comput. Methods Appl. Mech. Eng.* 344 (2019) 512–537.
- [14] Y. Wang, Z. Wang, Z. Xia, L.H. Poh, Structural design optimization using isogeometric analysis: a comprehensive review, *CMES Comp. Model Eng. Sci.* 117 (2018) 455–507.
- [15] X. Li, L. Zhang, B. Li, Heat transfer augmentation in microchannel heat sink based on isogeometric topology optimization framework, *Appl. Math. Model.* 104 (2021) 163–187.
- [16] J. Liu, A.T. Gaynor, S. Chen, Z. Kang, K. Suresh, A. Takezawa, L. Li, J. Kato, J. Tang, C.C.L. Wang, L. Cheng, X. Liang, A.C. To, Current and future trends in topology optimization for additive manufacturing, *Struct. Multidiscip. Optim.* 57 (2018) 2457–2483.

- [17] Y. Zheng, D. Da, H. Li, M. Xiao, L. Gao, Robust topology optimization for multi-material structures under interval uncertainty, *Appl. Math. Model.* 78 (2020) 627–647.
- [18] N.A. Kallioras, N.D. Lagaros, DL-SCALE: a novel deep learning-based model order upscaling scheme for solving topology optimization problems, *Neural Comput. Appl.* (2020).
- [19] S. Wang, E.d. Sturler, G.H. Paulino, Large-scale topology optimization using preconditioned Krylov subspace methods with recycling, *Int. J. Numer. Methods Eng.* 69 (2007) 2441–2468.
- [20] O. Amir, O. Sigmund, B.S. Lazarov, M. Schevenels, Efficient reanalysis techniques for robust topology optimization, *Comput. Methods Appl. Mech. Eng.* 245 (2012) 217–231.
- [21] O. Amir, N. Aage, B.S. Lazarov, On multigrid-CG for efficient topology optimization, *Struct. Multidiscip. Optim.* 49 (2014) 815–829.
- [22] C. Gogu, Improving the efficiency of large scale topology optimization through on-the-fly reduced order model construction, *Int. J. Numer. Methods Eng.* 101 (2015) 281–304.
- [23] P. Wei, M.Y. Wang, X. Xing, A study on X-FEM in continuum structural optimization using a level set model, *Comput. Aided Des.* 42 (2010) 708–719.
- [24] J. Zhang, W. Zhang, J. Zhu, L. Xia, Integrated layout design of multi-component systems using XFEM and analytical sensitivity analysis, *Comput. Methods Appl. Mech. Eng.* 245 (2012) 75–89.
- [25] Z. Liao, Y. Zhang, Y. Wang, W. Li, A triple acceleration method for topology optimization, *Struct. Multidiscip. Optim.* 60 (2019) 727–744.
- [26] W. Zheng, Y. Wang, Y. Zheng, D. Da, Efficient topology optimization based on DOF reduction and convergence acceleration methods, *Adv. Eng. Softw.* 149 (2020) 102890.
- [27] J.K. Guest, L.C. Smith Genut, Reducing dimensionality in topology optimization using adaptive design variable fields, *Int. J. Numer. Methods Eng.* 81 (2010) 1019–1045.
- [28] S.Y. Kim, I.Y. Kim, C.K. Mechefske, A new efficient convergence criterion for reducing computational expense in topology optimization: reducible design variable method, *Int. J. Numer. Methods Eng.* 90 (2012) 752–783.
- [29] Y. Wang, Z. Liao, M. Ye, Y. Zhang, W. Li, Z. Xia, An efficient isogeometric topology optimization using multilevel mesh, MGCG and local-update strategy, *Adv. Eng. Softw.* 139 (2020) 102733.
- [30] W. Li, P. Suryanarayana, G.H. Paulino, Accelerated fixed-point formulation of topology optimization: Application to compliance minimization problems, *Mech. Res. Commun.* 103 (2020) 103469.
- [31] Y. Yu, T. Hur, J. Jung, I.G. Jang, Deep learning for determining a near-optimal topological design without any iteration, *Struct. Multidiscip. Optim.* 59 (2019) 787–799.
- [32] A. Chandrasekhar, K. Suresh, TOuNN: Topology Optimization using Neural Networks, *Struct. Multidiscip. Optim.* (2020).
- [33] Z. Xia, Y. Wang, Q. Wang, C. Mei, GPU parallel strategy for parameterized LSM-based topology optimization using isogeometric analysis, *Struct. Multidiscip. Optim.* 56 (2017) 413–434.
- [34] J. Martínez-Frutos, P.J. Martínez-Castejón, D. Herrero-Pérez, Efficient topology optimization using GPU computing with multilevel granularity, *Adv. Eng. Softw.* 106 (2017) 47–62.
- [35] K. Maute, E. Ramm, Adaptive topology optimization, *Struct. Optim.* 10 (1995) 100–112.
- [36] G.C. DeRose Jr, A.R. Díaz, R. Alejandro, Hierarchical solution of large-scale three-dimensional topology optimization problems, in: *Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, 1996 V003T003A074.
- [37] C.Y. Lin, J.N. Chou, A two-stage approach for structural topology optimization, *Adv. Eng. Softw.* 30 (1999) 261–271.
- [38] J.C.A. Costa, M.K. Alves, Layout optimization with h-adaptivity of structures, *Int. J. Numer. Methods Eng.* 58 (2003) 83–102.
- [39] R. Stainko, An adaptive multilevel approach to the minimal compliance problem in topology optimization, *Commun. Numer. Methods Eng.* 22 (2006) 109–118.
- [40] T.H. Nguyen, G.H. Paulino, J. Song, C.H. Le, A computational paradigm for multiresolution topology optimization (MTO), *Struct. Multidiscip. Optim.* 41 (2010) 525–539.
- [41] Q.X. Lieu, J. Lee, A multi-resolution approach for multi-material topology optimization based on isogeometric analysis, *Comput. Methods Appl. Mech. Eng.* 323 (2017) 272–302.
- [42] Q.X. Lieu, J. Lee, Multiresolution topology optimization using isogeometric analysis, *Int. J. Numer. Methods Eng.* 112 (2017) 2025–2047.
- [43] M. Bruggi, M. Verani, A fully adaptive topology optimization algorithm with goal-oriented error control, *Comput. Struct.* 89 (2011) 1481–1493.
- [44] Y. Wang, Z. Kang, Q. He, An adaptive refinement approach for topology optimization based on separated density field description, *Comput. Struct.* 117 (2013) 10–22.
- [45] A. Nana, J.-C. Cuillière, V. Francois, Towards adaptive topology optimization, *Adv. Eng. Softw.* 100 (2016) 290–307.
- [46] K.N. Chau, K.N. Chau, T. Ngo, K. Hackl, H. Nguyen-Xuan, A polytree-based adaptive polygonal finite element method for multi-material topology optimization, *Comput. Meth. Appl. Mech. Eng.* 332 (2018) 712–739.
- [47] M.A. Salazar de Troya, D.A. Tortorelli, Adaptive mesh refinement in stress-constrained topology optimization, *Struct. Multidiscip. Optim.* 58 (2018) 2369–2386.
- [48] A. Vogel, P. Junker, Adaptive thermodynamic topology optimization, *Struct. Multidiscip. Optim.* 63 (2021) 95–119.
- [49] S. Wang, E. de Sturler, G.H. Paulino, Dynamic adaptive mesh refinement for topology optimization, *arXiv preprint arXiv:1009.4975*, (2010).
- [50] A.B. Lambé, A. Czekanski, Topology optimization using a continuous density field and adaptive mesh refinement, *Int. J. Numer. Methods Eng.* 113 (2018) 357–373.
- [51] T.-P. Fries, A. Byfut, A. Alizada, K.W. Cheng, A. Schröder, Hanging nodes and XFEM, *Int. J. Numer. Methods Eng.* 86 (2011) 404–430.
- [52] O. Sigmund, On the usefulness of non-gradient approaches in topology optimization, *Struct. Multidiscip. Optim.* 43 (2011) 589–596.
- [53] K. Liu, A. Tovar, An efficient 3D topology optimization code written in Matlab, *Struct. Multidiscip. Optim.* 50 (2014) 1175–1196.
- [54] Y. Wang, Z. Kang, Q. He, Adaptive topology optimization with independent error control for separated displacement and density fields, *Comput. Struct.* 135 (2014) 50–61.
- [55] K.J. Bathe, *Finite element procedures*, Klaus-Jürgen Bathe, 2006.
- [56] J.N. Reddy, *Introduction to the Finite Element Method*, McGraw-Hill Education, 2019.
- [57] X. Xie, S. Wang, Y. Wang, N. Jiang, W. Zhao, M. Xu, Truncated hierarchical B-spline-based topology optimization, *Struct. Multidiscip. Optim.* 62 (2020) 83–105.
- [58] E. Andreassen, A. Clausen, M. Schevenels, B.S. Lazarov, O. Sigmund, Efficient topology optimization in MATLAB using 88 lines of code, *Struct. Multidiscip. Optim.* 43 (2011) 1–16.
- [59] O. Sigmund, Morphology-based black and white filters for topology optimization, *Struct. Multidiscip. Optim.* 33 (2007) 401–424.
- [60] T. Yu, T.Q. Bui, Numerical simulation of 2-D weak and strong discontinuities by a novel approach based on XFEM with local mesh refinement, *Comput. Struct.* 196 (2018) 112–133.
- [61] Z. Wang, T. Yu, T.Q. Bui, S. Tanaka, C. Zhang, S. Hirose, J.L. Curiel-Sosa, 3-D local mesh refinement XFEM with variable-node hexahedron elements for extraction of stress intensity factors of straight and curved planar cracks, *Comput. Methods Appl. Mech. Eng.* 313 (2017) 375–405.