Advances in Engineering Software 149 (2020) 102890

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Efficient topology optimization based on DOF reduction and convergence acceleration methods

Wei Zheng[a], Yingjun Wang[a,*], Yongfeng Zheng[a], Daicong Da[b]

[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 Mechanical Engineering, University of Wisconsin-Madison, Madison, WI 53706, United States of America

ARTICLE INFO

Keywords:
Topology optimization
DOF reduction
Finite element equation
Convergence acceleration
Gray-scale suppression

ABSTRACT

This paper proposes a highly efficient topology optimization using two accelerated methods, which can reduce the degrees of freedom (DOFs) of the finite element equations and accelerate the iteration convergence of the topology optimization. For the DOF reduction, a method based on the empty elements and the displacement change during the topology iterations is presented to remove the DOFs from the finite element equations. For the convergence acceleration, a gray-scale suppression method is proposed to accelerate the polarization of design variables which accelerates the iteration convergence of the topology optimization. Three numerical examples including 2D and 3D cases are tested, and the results show that the proposed method can significantly improve the efficiency of the topology optimization and obtain the optimization results with the same accuracy. The computational time is only about 7% - 29% compared to the conventional topology optimization method.

1. Introduction

As one of the most promising techniques of structural optimization, the topology optimization dedicates to search the optimal material distribution within a specified design area, which has larger design freedom compared with traditional size optimization and shape optimization. In the past three decades, many topology optimization methods have been emerged such as homogenization method (HM) [1], solid isotropic material with penalization (SIMP) [2] and rational approximation of material properties (RAMP) [3], evolutionary structural optimization (ESO) [4,5] and bi-directional evolutionary structural optimization (BESO) [6–8], level set method [9,10] and isogeometric analysis [11–13]. These methods can be divided into two types: element description and boundary expression, and both of them have their own advantages in different applications. Topology optimization has been successfully used to solve numerous engineering problems such as minimum compliance [14], compliant mechanism synthesis [15], heat conduction [16], the design of composite structures [17], fluid structure interaction [18,19], microstructure optimization [20–22] and so on.

In general, the computational efficiency of the topology optimization cannot meet the actual requirements in practical, since the topology optimization problems are complex, which must be solved iteratively and the finite element analysis (FEA) is required in each iteration. Therefore, it is of significance to pursue efficient strategies to speed up the topology optimization fundamentally. Taking the efficient 88-line MATLAB code [23] as an example, the low efficiency mainly reflects in two aspects: one is that the solution of the finite element equations is slow, and the other is that the optimization process requires hundreds of iterations, and the FEA problem needs to be solved in each iteration. Therefore, speeding up the solution of the finite element equations and accelerating the convergence are two key issues to improve the efficiency of the topology optimization.

To solve the finite element equations, there are generally two types of methods: direct methods [24,25] such as Gaussian elimination and Cholesky decomposition, and iterative methods [26] such as Gauss-Seidel iteration method, successive over-relaxation (SOR) method, and some others based on krylov subspace like preconditioned conjugate gradient (PCG) method and generalized minimal residual (GMRES) algorithm. The computational complexity of direct methods is $O(n^3)$, while the iteration method is $O(n^2)$, whose computational work in one iteration is essentially the multiplication operation of matrix vectors, and the number of iterations required for convergence is much less than $O(n)$ if a suitable preconditioner is selected. Thus, for solving lager linear equations, the iteration methods perform better than the direct methods in computational time, meanwhile it uses less memory and is

* Corresponding author.
E-mail address: wangyj84@scut.edu.cn (Y. Wang).

https://doi.org/10.1016/j.advengsoft.2020.102890
Received 22 April 2020; Received in revised form 27 June 2020; Accepted 3 August 2020
Available online 14 August 2020

easier to be parallelized [27]. The computational time of the iteration methods will be further reduced if the DOFs of the equations are reduced. Many fruitful methods based on the DOF reduction of the finite element equations have been applied to topology optimizations. Amir et al. [28] used the approximate reanalysis method in the topology optimization to reduce the computational time of FEA, which follows the combined approximations (CA) approach [29] that was initially proposed by Kirsch [30]. Furthermore, Amir [31] indicated that approximate reanalysis is more efficient to solve the minimum weight problem than the common minimum compliance problem when integrated with recycled preconditioning and Long et al. [32] proposed a novel minimum weight formulation of topology optimization using reanalysis approach. The approximate reanalysis method converts the solution of the finite element equations into a linear combination of several basis vectors. Although the method greatly reduces the DOFs of the finite element equations, it requires lots of mathematical operations including matrix factorization, constructing reduced basis, and the Gram-Schmidt orthogonalization, which increases the computational complexity. Gogu [33] proposed a reduced order model (ROM) construction method which dynamically constructed the reduced bases using previous solutions of the finite element equations, but the convergence is influenced by the ROM method. Liao et al. [34] solved the finite element equations based on multilevel mesh, which can reduce the DOFs of the finite element equations in the coarse mesh stage of the topology optimization. More methods about reducing the DOFs of the finite element equations can be found in [35,36].

In terms of accelerating the convergence of the topology optimization, reducing some design variables is a good choice than using all design variables [37]. Kim et al. [38] proposed a reducible design variable method that reduced some design variables which are quickly converged so as to accelerate the convergence of the topology optimization. Liao et al. [34] proposed a local-update strategy that only updated the solid region and its neighbor boundary region. Apart from reducing design variables to speed up the convergence, there are still some other methods. Zhao et al. [39] proposed proportional and differential OC method by combining the fixed-point iteration method with differential control theory, which converged faster than the standard OC method. Okamoto et al. [40] adopted the method of moving asymptotes for the topology optimization based on level set function in 3D magnetic field system and found the convergence was better than the conventional level set method.

Although many methods related to the topology optimization acceleration have been already proposed, the highly efficient and accurate methods are still worth researching. This paper focuses on the DOF reduction of the finite element equations and the convergence acceleration of the topology optimization. To reduce the DOFs of the finite element equations, this paper proposes a method reducing the DOFs of the finite element equations in two steps. The DOFs are firstly reduced by avoiding calculating the displacements of the nodes surrounded by empty elements, and then further reduced by avoiding calculating the displacements that have little change after one iteration. To accelerate the convergence of the topology optimization, this paper proposes a gray-scale suppression method based on the research [41] to accelerate the polarization of design variables which can significantly speed up the convergence.

The remainder of this paper is organized as follows. Section 2 reviews the theory of the topology optimization including the powerful SIMP, OC algorithm and some knowledge about finite element method (FEM). Section 3 introduces the accelerated methods including both the DOF reduction and the convergence acceleration. Section 4 presents the flowchart of algorithm implementation. Numerical examples are showed in Section 5, and Section 6 draws the conclusion.

2. Theory background

This section introduces the topology optimization based on SIMP using OC method and some knowledge about FEA. For more details about these theories, please refer to [23,42–44].

2.1. Topology optimization based on SIMP

For SIMP method, the design domain is discretized into numerous finite elements and each element is assigned a density between 0 and 1. The element density x_e determines the Young's modulus E_e as follows:

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

where E_0 is the elastic modulus of the material assigned to the solid element whose density is 1, $E_{\min}$ is a very small elastic modulus assigned to the void element whose density is 0 for the purpose of preventing the stiffness matrix becoming singular, and p is a penalization factor (typically $p = 3$) introduced to push element density toward 0 or 1.

The classical minimum compliance problem of the topology optimization can be written as follows:

$$\begin{aligned} &\text{Find:}\mathbf{x} = (\ x_1,\ x_2,\ x_3\cdots,x_N)^{\mathrm{T}} \\ &\text{Min}:c(\mathbf{x}) = \mathbf{U}^{\mathrm{T}}\mathbf{K}\mathbf{U} = \sum_{e=1}^{N} E_e(x_e)\mathbf{u}_e^{\mathrm{T}}\mathbf{k}_e\mathbf{u}_e \\ &\text{s. t.}\ :\begin{cases} V(\mathbf{x})/V_0 \le f \\ \mathbf{K}\mathbf{U} = \mathbf{F} \\ 0 \le x \le 1 \end{cases} \end{aligned} \tag{2}$$

where $\boldsymbol{x} = (x_1,x_2,x_3,\ldots,\ x_N)^{\mathrm{T}}$ is the design variable vector; c is the structural compliance. $\boldsymbol{U}$ is the global displacement vector; $\boldsymbol{F}$ is the global force vector; $\boldsymbol{K}$ is the global stiffness matrix; N is the number of the elements; $\boldsymbol{u}_e$ is the element displacement vector; $\boldsymbol{k}_e$ is the element stiffness matrix with unit Young's modulus; $V(\boldsymbol{x})$ and V_0 are the material volume and design volume, and f is the prescribed volume fraction.

To solve the optimization problem proposed above, the OC method is utilized, which is a heuristic scheme and can be written as follows:

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

where m is a positive move limit, η is a numerical damping coefficient (typically $\eta = 1/2$), and 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 λ is the Lagrangian multiplier introduced to satisfy the volume constraint, which can be found by means of a bisection algorithm.

The sensitivities of the objective function and the material volume V with respect to the element density x_e are:

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

$$\frac{\partial V}{\partial x_e} = 1 \tag{6}$$

To overcome the numerical problems including checkboard, mesh-dependences and local minima in the topology optimization, the sensitivity filter is adopted in this work, and more details about filter methods can be referred to [45].

2.2. Finite element analysis

This section discusses the basic knowledge about FEA especially the assembly of the global stiffness matrix for the purpose of a better understanding to the DOF reduction presented in Section 3.1.

In FEA, the stiffness matrix $\boldsymbol{k}_e$ for the e-th element is proportional to the Young's modulus E_e and can be deduced by the Eq. (1). When the stiffness matrix for each element is known, and the problem to be solved is to assemble the global stiffness matrix. As mentioned in

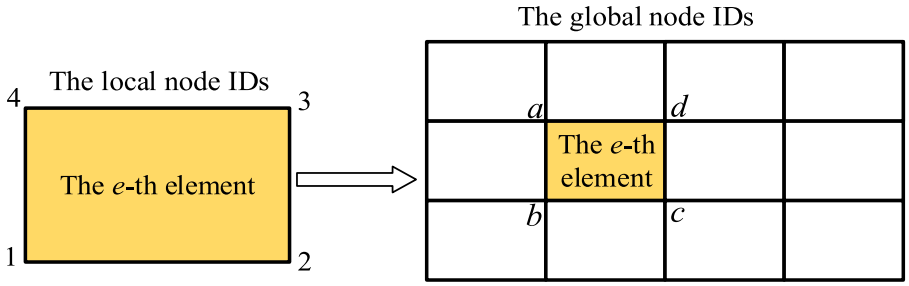

Fig. 1. The local node IDs and the global node IDs for the *e*-th element.

Table 1
The local DOF indices and the global DOF indices corresponding to the node IDs

Local node IDs	Local DOF indices	Global node IDs	Global DOF indices
1	1, 2	*b*	2*b*-1, 2*b*
2	3, 4	*c*	2*c*-1, 2*c*
3	5, 6	d	2*d*-1, 2*d*
4	7, 8	*a*	2*a*-1, 2*a*

Section 2.1, the design domain is discretized into finite elements. Take the linear rectangular element as an example to discuss the assembly of the global stiffness matrix. Every rectangular element has four nodes with eight DOFs. Each node is identified by node ID and each DOF has its index (DOF index). As shown in Fig. 1, for the *e*-th element, the local node IDs are 1, 2, 3 and 4, and the global node IDs are assumed to be *a, b, c* and *d*. The local DOF indices and the global DOF indices corresponding to the node IDs are shown in Table 1.

To convert the stiffness matrix from the local DOF indices to the global DOF indices, a transfer matrix should be introduced. For the *e*-th element, the form of transfer matrix G_e is expressed as follows:

$$G_e = \begin{array}{c} \begin{matrix} 1 & 2 & \dots & 2a-1 & 2a & 2b-1 & 2b & \dots & 2c-1 & 2c & 2d-1 & 2d & \dots & 2n-1 & 2n \end{matrix} \\ \begin{bmatrix} 0 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & 1 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 1 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{matrix} \end{array} \tag{7}$$

Where n is the number of nodes and G_e is an $8 \times 2n$ matrix consisting of only 0 or 1 elements. According to the correspondence between the local DOF indices and the global DOF indices, the positions of 1 or 0 elements in the transfer matrix can be found. As shown in Table 1, the local DOF index 1 corresponds to the global DOF index 2*b*-1, 2 corresponds to 2*b*, 3 corresponds to 2*c*-1, 4 corresponds to 2*c* and in the same way, the last is 8 corresponds to 2*a*. The row 1 and column 2*b*-1, row 2 and column 2*b*, row 3 and column 2*c*-1, row 4 and column 2*c* and so on are exactly the positions of 1 elements in the transfer matrix. Except that the elements on the eight positions are equal to 1, all elements in the transfer matrix are equal to 0. The element stiffness matrix $\widetilde{\mathbf{k}}_e$ in the global DOF indices is expressed as:

$$\widetilde{\mathbf{k}}_{\mathbf{e}} = \mathbf{G}_e^T \mathbf{k}_e \mathbf{G}_e = \frac{E_{\min} + x_e^p (E_0 - E_{\min})}{E_0} \mathbf{G}_e^T \mathbf{k}_0 \mathbf{G}_e \tag{8}$$

The global stiffness matrix K is:

$$\mathbf{K} = \sum_{e=1}^{N} \widetilde{\mathbf{k}}_e \tag{9}$$

The final task of FEA is solving the equations to get the nodal displacements.

$$\mathbf{KU} = \mathbf{F} \tag{10}$$

3. Acceleration methods

This paper proposes two acceleration schemes to improve the efficiency of the topology optimization. One is the DOF reduction of the finite element equations, and the other is the convergence acceleration during the iterative process.

3.1. The DOF reduction of the finite element equations

3.1.1. The DOF reduction based on empty elements

In the topology optimization of geometrically nonlinear structures, Buhl et al. [46] found that displacements in the nodes surrounded by empty elements kept oscillating and proposed to remove such nodes in the convergence criterion of the Newton-Raphson iterations to circumvent the nonconvergence of iterations. In our work, what is different from the above-mentioned work is that we found that displacements in the nodes surrounded by empty elements have negligible influence on the process of topology optimization in linear structures and proposed to remove the DOFs of such nodes to improve the efficiency of FEA.

The traditional methods to solve the finite element equations consider all DOFs of the structure. As the topology optimization models are becoming more and more complex with increasing DOFs, the solution of the finite element equations is time-consuming. In fact, the DOFs can be reduced to improve the solving efficiency, i.e., some DOFs can be removed for solving the finite element equations.

The key issue is to determine which DOFs are not necessary in the solution. Take the 2D problem as an example, the design domain is discretized into 12 rectangular elements with 20 nodes and 40 DOFs as shown in Fig. 2.

In Fig. 2, the numbers in the center of the rectangle are the element IDs, next to the nodes are the node IDs, and in the brackets are the DOF indices. The finite element equations of the design domain can be expressed as:

$$\begin{bmatrix} K_{1,1} & K_{1,2} & \cdots & K_{1,39} & K_{1,40} \\ K_{2,1} & K_{2,2} & \cdots & K_{2,39} & K_{2,40} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ K_{39,1} & K_{39,2} & \cdots & K_{39,39} & K_{39,40} \\ K_{40,1} & K_{40,2} & \cdots & K_{40,39} & K_{40,40} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_{39} \\ U_{40} \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_{39} \\ F_{40} \end{bmatrix} \tag{11}$$

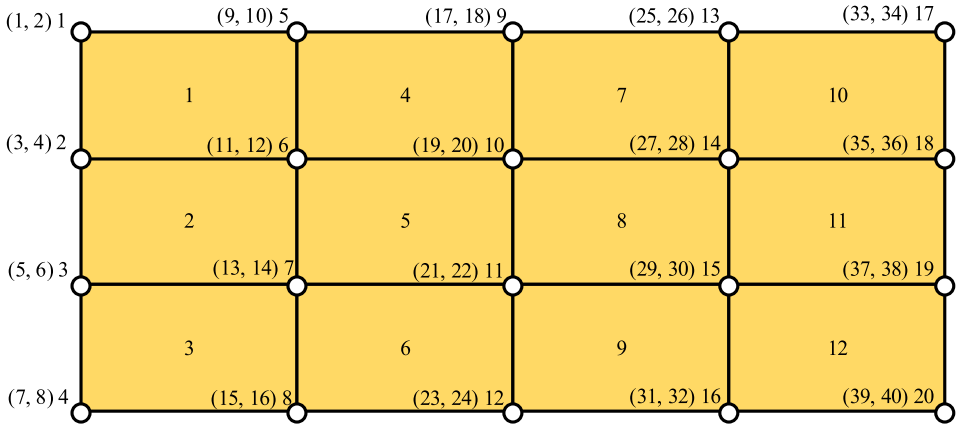

Fig. 2. The design domain is discretized into 12 rectangular elements with 20 nodes and 40 DOFs.

The subscripts of $K_{i,j}$, U_i, F_i ($i, j = 1, 2, 3, \ldots, 39, 40$) represent the i-th row and j-th column element in the global stiffness matrix K, the i-th element in the displacement vector U and the i-th element in the force vector F, respectively. For the topology optimization based on SIMP, the element density will be updated toward 0 or 1 as the iterations performing. Assuming that the densities of elements 2, 3, 5 and 6 are updated to 0 after one iteration as the white elements show in Fig. 3.

Ignoring the very small elastic modulus assigned to the empty elements, the element stiffness matrices of the elements 2, 3, 5 and 6 can be regarded as zero matrices. From Eqs. (8) and (9), i.e. the assembly process of the global stiffness matrix, it can be deduced that all elements in 13-th, 14-th rows and columns of the global stiffness matrix are equal to 0. It should be noted that for the boundary nodal DOF indices 5, 6, 7, 8, 15 and 16, all elements in their corresponding rows and columns of the global stiffness matrix are also equal to 0. The general situation is that if a node is surrounded by the empty elements like the red nodes shown in Fig. 3, the rows and columns of the global stiffness matrix corresponding to the DOF indices of the node should be equal to 0. Now just consider the 13-th, 14-th rows and columns of the global stiffness matrix, Eq. (11) can be written as:

$$\begin{bmatrix} \ddots & \cdots & 0 & 0 & \cdots & \cdots \\ \vdots & \ddots & 0 & 0 & \cdots & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & 0 & 0 & \ddots & \cdots \\ \vdots & \vdots & 0 & 0 & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \vdots \\ U_{13} \\ U_{14} \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \vdots \\ F_{13} \\ F_{14} \\ \vdots \\ \vdots \end{bmatrix} \tag{12}$$

We can remove 13-th, 14-th rows and columns of the global stiffness matrix, the corresponding displacements and forces in the displacement vector and the force vector. After this operation, new finite element equations with fewer DOFs are formed. The displacements solved by the new equations are the same as that solved by the original equations. The problem that comes with the operation is that the values of U_{13}, U_{14} are unknown. According to Eq. (5), the sensitivities of the elements 2, 3, 5 and 6 are equal to 0 regardless of the values of nodal displacements. That means U_{13}, U_{14} do not affect the values of sensitivities of the elements, and therefore do not affect the update of the design variables. Therefore, the unknown U_{13}, U_{14} will not influence on the optimization results. In the same way, U_5, U_6, U_7, U_8, U_{15} and U_{16} can also get rid of being calculated. During the iteration progress, more elements become empty means that more DOFs can be reduced.

The key problem of the method is how to find out all nodes surrounded by empty elements. To solve the problem, the nodal density is introduced. Fig. 4 shows the relationship between the element density and the nodal density. For each element, the element density is divided into 4 identical densities, which are assigned to the four nodes on the element, e.g., the density of element 1 is divided into 4 identical densities which are assigned to the nodes 1, 2, 5 and 6. If a node is shared by multiple elements, the nodal density is equal to average density of the shared elements, e.g., nodal density of node 2 is equal to the average density of elements 1 and 2, nodal density of 6 is equal to the average density of elements 1, 2, 4 and 5.

The mapping relationship between nodes and elements of the 6

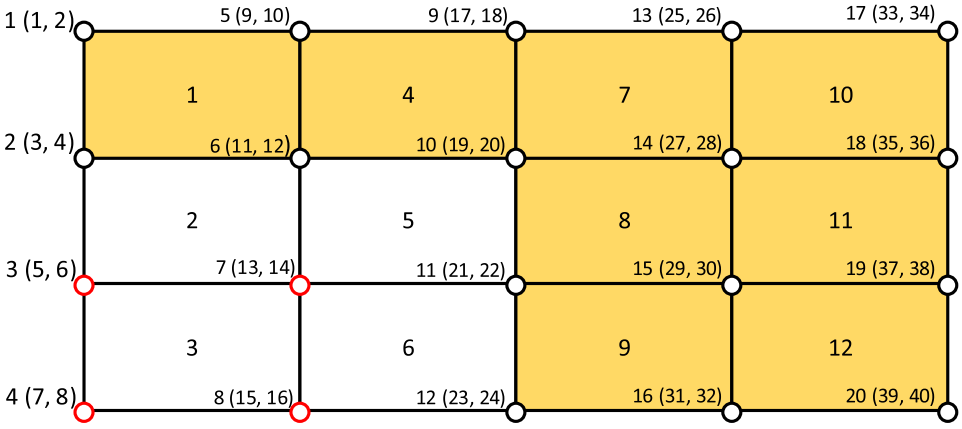

Fig. 3. The densities of elements 2, 3, 5 and 6 are updated to 0 after one iteration.

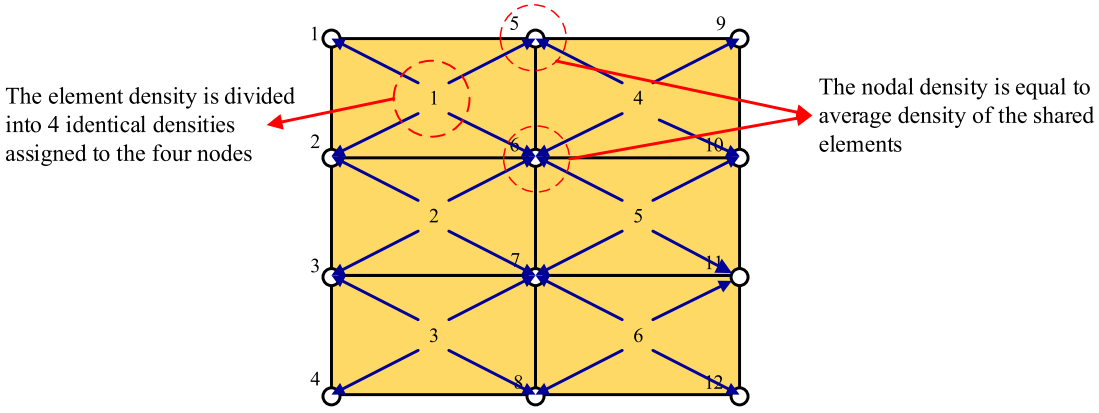

Fig. 4. The relationship between element density and nodal density.

elements in Fig. 4 is formulated as follows:

$$\text{node IDs}=\begin{bmatrix}1&2&5&6\\2&3&6&7\\3&4&7&8\\5&6&9&10\\6&7&10&11\\7&8&11&12\end{bmatrix}\begin{matrix}\leftarrow\text{element 1}\\\leftarrow\text{element 2}\\\leftarrow\text{element 3}\\\leftarrow\text{element 4}\\\leftarrow\text{element 5}\\\leftarrow\text{element 6}\end{matrix}$$

$$\text{element IDs}=\begin{bmatrix}1&0&0&0\\1&2&0&0\\2&3&0&0\\3&0&0&0\\1&4&0&0\\1&2&4&5\\\vdots&\vdots&\vdots&\vdots\\6&0&0&0\end{bmatrix}\begin{matrix}\leftarrow\text{node 1}\\\leftarrow\text{node 2}\\\leftarrow\text{node 3}\\\leftarrow\text{node 4}\\\leftarrow\text{node 5}\\\leftarrow\text{node 6}\\\vdots\\\leftarrow\text{node 12}\end{matrix} \tag{13}$$

In this way, the nodal density for each node can be obtained. The DOFs corresponding to the nodes with 0 nodal density are what we need to find, which can be removed to improve the efficiency of the finite element equation solution.

3.1.2. DOF reduction based on the displacement change

Section 3.1.1 discussed the DOF reduction based on the empty elements. Some specific DOFs can be reduced to improve the efficiency of the finite element equation solution. This section discusses the DOF reduction based on the displacement change which further reduces the DOFs of the finite element equations.

Fig. 5 shows the curves of several displacement changes with the iteration number, which is tested by the MBB beam using the top88 code [23] in the case of 100 × 50 elements.

It can be seen from Fig. 5 that the displacements are approximate

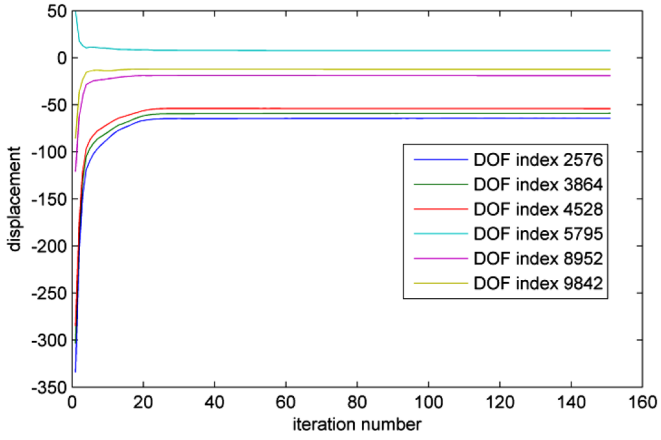

Fig. 5. The curves of several displacement changes with iteration number.

equal in the late stage of the topology optimization. Inspired by this phenomenon, it is supposed that some displacements do not need to be calculated if the displacements have very little change after a certain number of iterations, and therefore the DOFs can be further reduced.

If a certain displacement dose not change after a certain number of iterations, how to reduce the DOFs of the finite element equations is the key issue. Giving a global stiffness matrix as an 8 × 8 matrix for an example and the equations can be written as:

$$\begin{bmatrix}K_{1,1}&K_{1,2}&\cdots&K_{1,7}&K_{1,8}\\K_{2,1}&K_{2,2}&\cdots&K_{2,7}&K_{2,8}\\\vdots&\vdots&\ddots&\vdots&\vdots\\K_{7,1}&K_{7,2}&\cdots&K_{7,7}&K_{7,8}\\K_{8,1}&K_{8,2}&\cdots&K_{8,7}&K_{8,8}\end{bmatrix}_{8\times 8}\begin{bmatrix}U_1\\U_2\\\vdots\\U_7\\U_8\end{bmatrix}=\begin{bmatrix}F_1\\F_2\\\vdots\\F_7\\F_8\end{bmatrix} \tag{14}$$

Where the displacement U_2 is assumed to be almost unchanged, then the equations can be transformed as follows:

$$\begin{bmatrix}K_{1,1}&K_{1,3}&\cdots&K_{1,7}&K_{1,8}\\K_{3,1}&K_{3,3}&\cdots&K_{3,7}&K_{3,8}\\\vdots&\vdots&\ddots&\vdots&\vdots\\K_{7,1}&K_{7,3}&\cdots&K_{7,7}&K_{7,8}\\K_{8,1}&K_{8,3}&\cdots&K_{8,7}&K_{8,8}\end{bmatrix}_{7\times 7}\begin{bmatrix}U_1\\U_3\\\vdots\\U_7\\U_8\end{bmatrix}=\begin{bmatrix}F_1-K_{1,2}U_2\\F_3-K_{3,2}U_2\\\vdots\\F_7-K_{7,2}U_2\\F_8-K_{8,2}U_2\end{bmatrix} \tag{15}$$

where the 2-nd row and column of the global stiffness matrix and 2-nd row of the displacement vector and force vector have been removed. In order to ensure the displacements solved by the new equations are the same as that solve by the original equations, according to the matrix multiplication rules, the force vector should subtract the product of the vector in the 2-nd column of the stiffness matrix and U_2. Thus, new finite element equations with one DOF reduced is generated. In this way, more DOFs can be reduced if more displacements are unchanged after a certain number of iterations.

To find out the displacements that are unchanged. The following formula is utilized to describe the magnitude of displacement change quantitatively.

$$dis\_ch=\frac{\left|\sum_{i=1}^{M}\left(U^{(k-i+1)}-U^{(k-M-i+1)}\right)\right|}{\left|\sum_{i=1}^{M}U^{(k-i+1)}\right|} \tag{16}$$

where dis_ch represents the displacement change, k is the current iteration number, $U^{(k)}$ represents the displacement at the k-th iteration, and M is an integer set to 5 in this work. Then whether a displacement is almost unchanged or not can be decided by the following criterion: if dis_ch is less than a small threshold ε, the displacement is considered as unchanged.

For the reason that the small-change displacements are considered constant in the next iteration, this method introduces errors while

reducing the DOFs. As the iteration progresses, the errors may be very large that have great influence on the final optimized result. To control the error within an acceptable range, the following formula is adopted to quantify it.

$$err = \frac{||\mathbf{K}^*\mathbf{U}^* - \mathbf{F}^*||_2}{||\mathbf{F}^*||_2} \tag{17}$$

Where *err* represents the error, $\boldsymbol{K}^*$, $\boldsymbol{F}^*$ are the reduced global stiffness matrix and force vector, $\boldsymbol{U}^*$ is the solution solved by the new equations. In general, $\boldsymbol{K}^*\boldsymbol{U}^*$ is not totally equal to $\boldsymbol{F}^*$. If *err* is bigger than a tolerance *tol*, the DOFs will not be reduced to ensure the error is always smaller than the tolerance.

3.2. Convergence acceleration

In this section, two accelerated schemes are proposed to improve the convergence: one is based on reducing design variables, and the other is based on gray-scale suppression. The efficiency will be compared in the numerical examples presented in Section 5.

3.2.1. Convergence acceleration based on reducing design variables

Inspired by the research of [34,38,47], a variable-reduction scheme is proposed to accelerate convergence based on the convergence criterion that the maximum difference of all element densities between two successive iterations are less than 0.01.

Fig. 6 shows the curves of some random-select element density changes with iteration number in two different cases tested by the MBB beam using the top88 code. It can be seen that some element densities keep stable around 0 or 1. Therefore, it's not necessary to update these design variables.

A criterion to determine whether an element density needs to be updated is shown as follows:

$$\begin{aligned}&\max(x^{(k)},\ x^{(k-1)},\ \cdots, x^{(k-Z+2)},\ x^{(k-Z+1)}) < 0.01\\ &\text{or } \min(x^{(k)},\ x^{(k-1)},\ \cdots, x^{(k-Z+2)},\ x^{(k-Z+1)}) > 0.99\end{aligned} \tag{18}$$

where k is the current iteration number, $x^{(k)}$ represent the density at the k-th iteration, and Z is an integer set to 5 in this work. If the density of an element has a maximum value less than 0.01 or a minimum value greater than 0.99 for Z successive iterations, this element density will not be updated. For the densities that are not updated, they will be assigned to 0 for less than 0.01 or to 1 for greater than 0.99. To guarantee the accuracy of the optimization results, the variable-reduction scheme is not adopted in the early stage of topology optimization for the reason that some densities may evolve toward intermediate value from 0 or 1 instead of keeping stable around 0 or 1 as the curves of element IDs 978 and 2001 show in Fig. 6(b).

The relative variation of the objective function value is utilized to determine whether it is in the early stage of the topology optimization, the calculation formula can be expressed as follows:

$$obj\_ch = \frac{\left|\sum_{i=1}^{X}(c^{(k-i+1)} - c^{(k-X-i+1)})\right|}{\left|\sum_{i=1}^{X} c^{(k-i+1)}\right|} \tag{19}$$

where *obj_ch* represents the objective change, k is the current iteration number, $c^{(k)}$ represent the k-th iteration compliance, and X is an integer set to 5 in this work. If *obj_ch* is bigger than a threshold δ, the current stage is regarded as the early stage of the topology optimization. The design variables can start to be reduced when *obj_ch* is smaller than δ. Different values of δ will have different effects on the optimized result, which will be discussed in Section 4.3.

3.2.1. Convergence acceleration based on gray-scale suppression

Fig. 7 shows the density distribution of the final structure of the MBB beam optimized by the top88 code in the case of 100 × 50 elements. Although the OC method push the intermediate densities toward 0 or 1, there are still many intermediate densities in the final optimized result as shown in Fig. 7. The convergence becomes very slow because some intermediate densities cannot approximately equal to 0 or 1 in a stable way.

According to this phenomenon, we use a function to continuously push the intermediate density towards 0 or 1 after being updated by OC. This function is shown in Eq. (20) which is obtained by moving the well-known sigmoid function 0.5 units in the positive direction of x-axis as shown in Fig. 8.

$$x'_{\text{new}} = \frac{1}{1 + e^{-a(x_{\text{new}}-0.5)}} \tag{20}$$

where x_{new} is the density updated by OC given by Eq. (3), x'_{new} is the final updated density, a is an parameter determining the degree of the push. The function curves with different values of a is shown in Fig. 9.

It should be noted that the parameter a is changeable during the topology optimization progress. In this work, a adopts an inverse proportional function:

$$a = \frac{t}{obj\_ch} \tag{21}$$

where t is a constant. During the topology optimization progress, the objective change *obj_ch* gradually decrease, a gradually increase, and the degree of pushing gradually increase to accelerate the convergence as much as possible. If t is too small, the high efficiency of the convergence acceleration cannot be fully obtained. If t is too large, the accuracy of the optimized results may be influenced. In this work, we

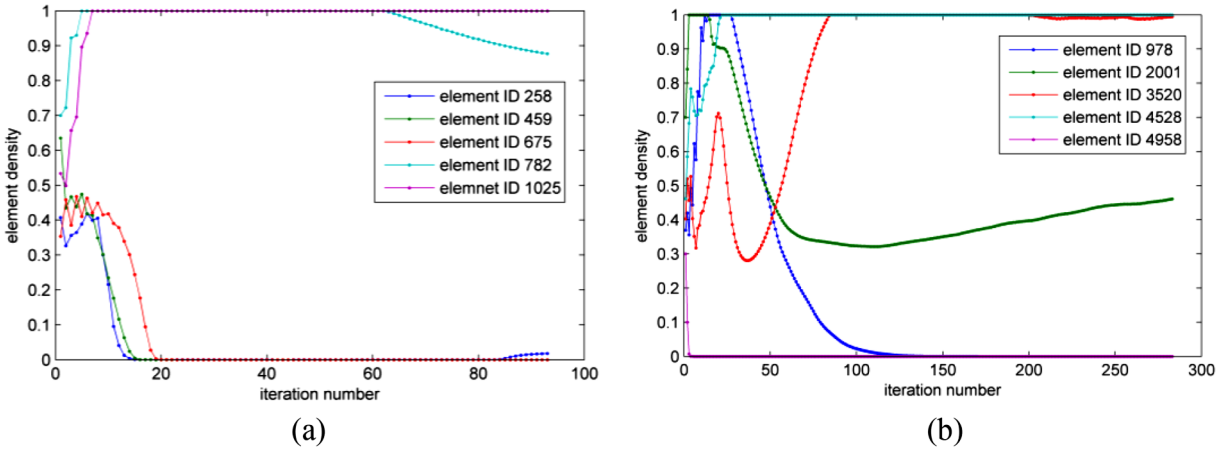

Fig. 6. Element density changes with iteration number. (a) the case of 60 × 30 elements, (b) the case of 100 × 50 elements.

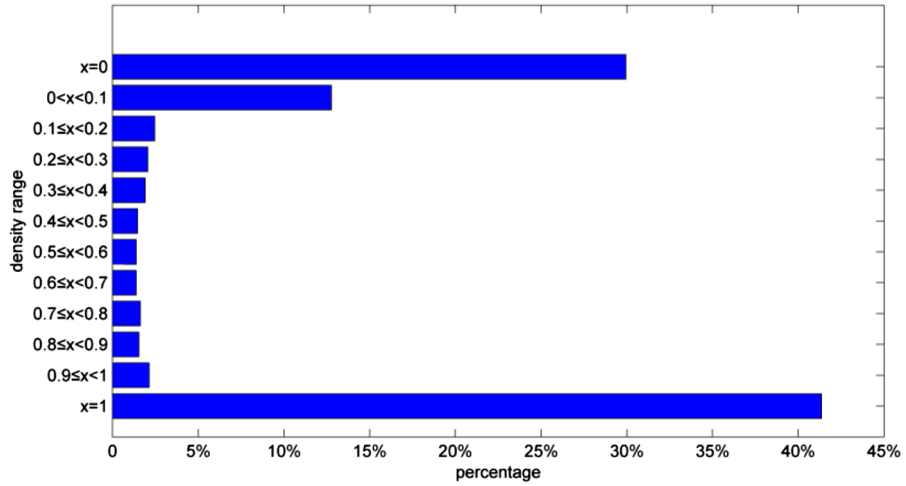

Fig. 7. The density distribution of the final structure of the MBB beam.

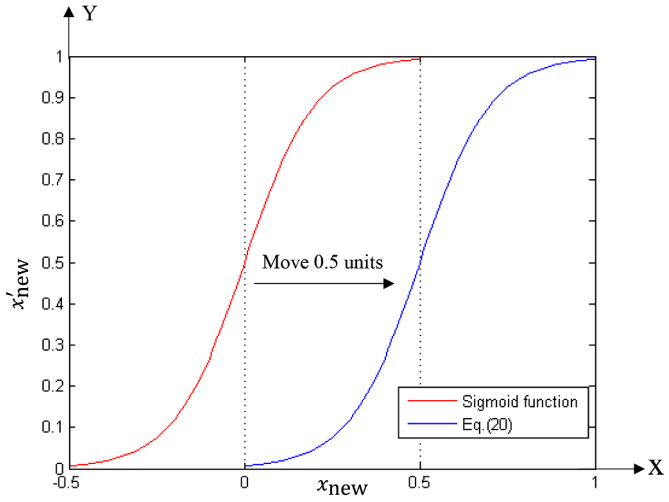

Fig. 8. The relationship between the sigmoid function and Eq. (20).

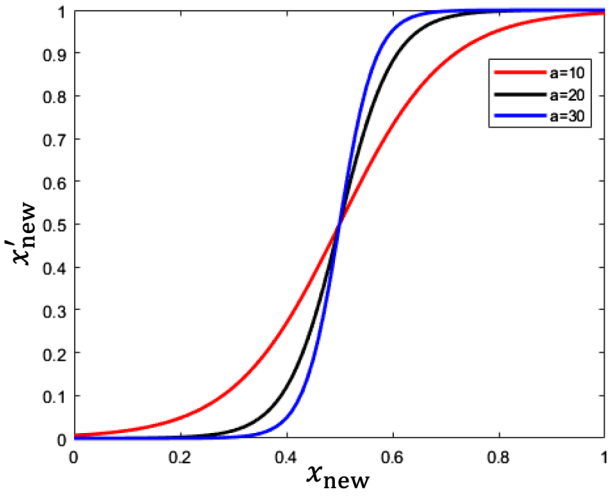

Fig. 9. The function curves with different parameters *a*.

set *t* to 0.6. The function should not be used in the early stage of the topology optimization since the structure of the topology optimization in the early stage is not stable. The identification of the early stage can be referred to the Eq. (19).

Table 2
Algorithm 1

Algorithm 1 DOF reduction based on the empty elements and the displacement change
Given the displacement change threshold ε, the error tolerance *tol*
Initialize the error *err* to 0
Calculate nodal densities
Find out node IDs with 0 nodal density
Calculate the DOF indices for the node IDs with 0 nodal density
Reduce the DOFs according to the DOF indices
if *err* < *tol* then
  Calculate the displacement change *dis_ch* by Eq. (16)
  Find the indices of the displacements whose changes are less than ε
  Reduce the DOFs according to the indices of the displacements
  Solve the reduced equations for unknown $\boldsymbol{U}$
  Calculate *err* by Eq. (17)
else
  Solve the equations formed in step 6 for $\boldsymbol{U}$
end if
Return $\boldsymbol{U}$

Table 3
Algorithm 2

Algorithm 2 Convergence acceleration based on reducing design variables
Given the objective change threshold δ
Calculate objective change *obj_ch* by Eq. (19)
if *obj_ch* < δ
  Find the densities that are less than 0.01 or bigger than 0.99 for 5 successive iterations and assign 0 or 1 to them.
  Reduce the design variables according to the found densities.
  Update the reduced design variables with OC
else
  Update all design variables with OC
end if
Return the design variables

Table 4
Algorithm 3

Algorithm 3 Convergence acceleration based on gray-scale suppression
Given the objective change threshold δ
Calculate objective change *obj_ch* by Eq. (19)
if *obj_ch* < δ
  Update design variables with OC to get x_{new}
  Calculate x'_{new} by Eq. (20)
else
  Update design variables with OC
end if
Return design variables

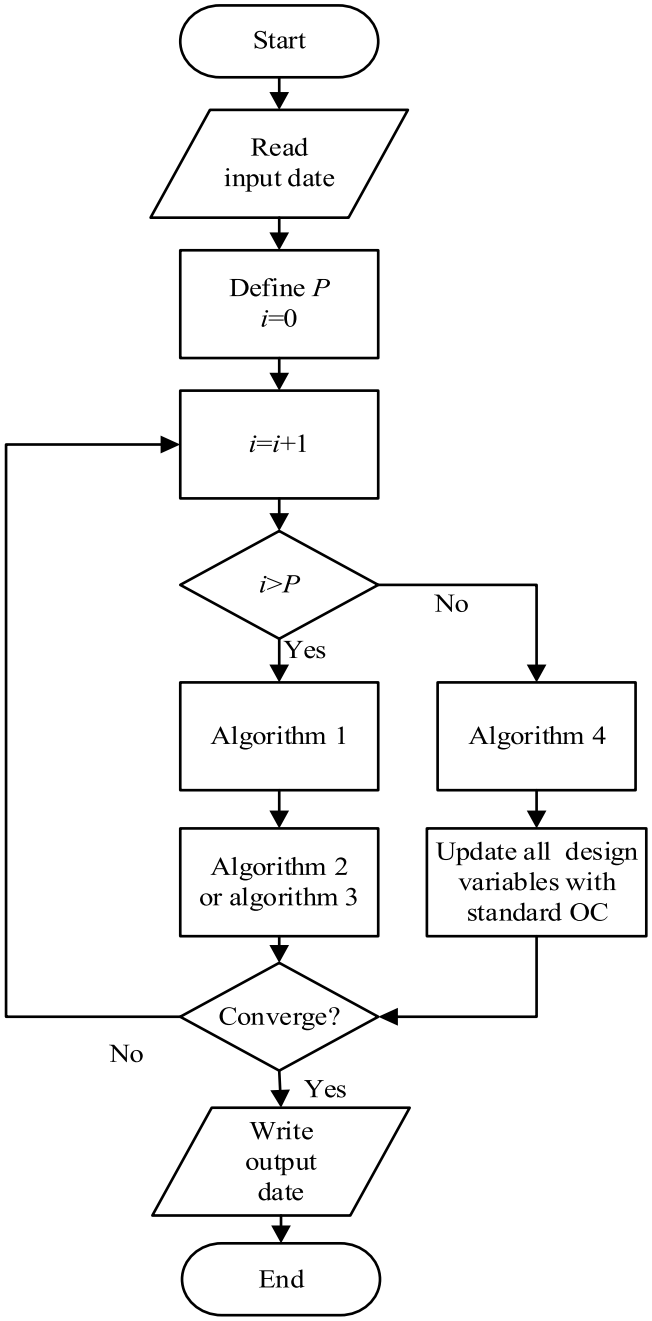

Fig. 10. The flowchart of the whole algorithm

Table 5
Algorithm 4

Algorithm 4 DOF reduction based on the empty elements
1 Calculate the nodal densities
2 Find out the node IDs with 0 nodal density
3 Calculate the DOF indices for the node IDs with 0 nodal density
4 Reduce the DOFs according to the DOF indices
5 Solve the reduced equations
6 Return $\boldsymbol{U}$

It can be seen from Fig. 9 that the intermediate densities updated with the gray-scale suppression method will continue to be pushed toward 0 or 1, which accelerate the polarization of design variables. The densities of design variables are stably polarized, and therefore the convergence speed will be greatly improved.

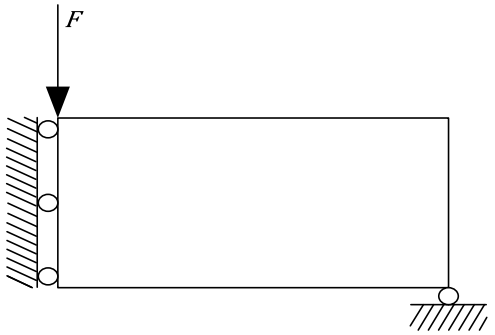

Fig. 11. The external load, design domain and boundary conditions of the half MBB beam example.

4. Algorithm implementation

4.1. Algorithm implementation for DOF reduction

Section 3.1 discussed two different methods to reduce the DOFs of the finite element equations: one is based on the empty elements, and the other is based on the displacement change. In this work, the two methods are integrated to further reduce the DOFs. The algorithm implementation is explained in Algorithm 1 (see Table 2).

4.2. Algorithm implementation for convergence acceleration

Two schemes to accelerate the convergence have been proposed in Section 3.2: one is reducing design variables, and the other is gray-scale suppression. The algorithm implementation of the two schemes are shown in Algorithm 2 (see Table 3) and Algorithm 3 (see Table 4), respectively.

4.3. The whole acceleration algorithm implementation

This section discusses the algorithm implementation of the combination of DOF reduction and convergence acceleration. In this work, during the early iterations, the gray-scale suppression and DOF reduction based on displacement change are not adopted since Eqs. (16), (18) and (19) require information from previous iterations. The number of the early iterations is symbolized as P, and then the flowchart of the whole program is shown in Fig. 10. The process of Algorithm 4 mentioned in Fig. 10 is shown in Table 5.

5. Numerical examples

In this section, three different numerical examples, i.e. MBB beam, 3D cantilever beam and compliant mechanism, will be presented to demonstrate the high computational efficiency of the acceleration algorithm proposed above.

In Section 5.1, MBB beam is presented to demonstrate the computational efficiency of the acceleration algorithm from its two accelerated aspects: DOF reduction and convergence acceleration. The proposed method is compared with the efficient top88 code. In Section 5.2, a 3D cantilever beam will be shown to demonstrate the high efficiency for 3D cases and study the specific effects of the objective change threshold δ on the acceleration algorithm. Finally, in Section 5.3, the compliant mechanism synthesis example will be shown to further prove the availability and high computational efficiency of the acceleration algorithm.

5.1. MBB beam

The MBB beam is the representative example in the topology optimization [23,44,45]. Taking advantage of structural symmetry and

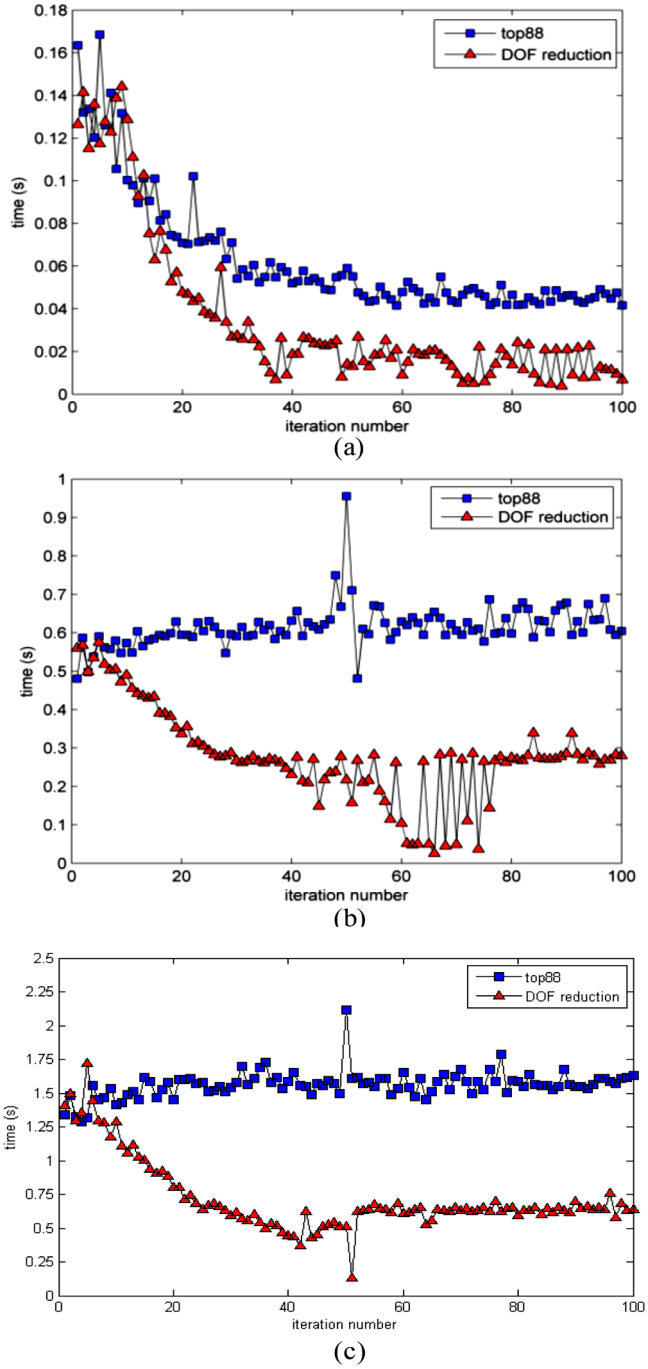

Fig. 12. Solution time comparison between top88 and DOF reduction on different meshes. (a) mesh with 100 × 50. (b) mesh with 200 × 100. (c) mesh with 300 × 150.

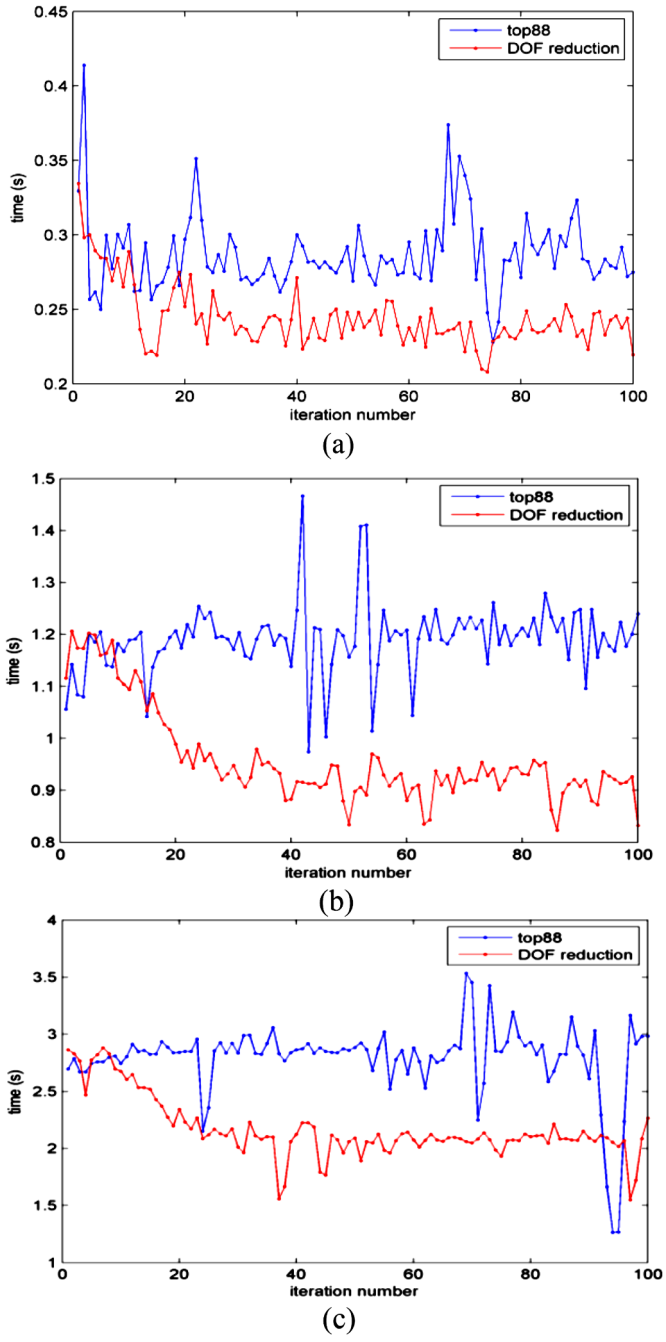

Fig. 13. Total time comparison between top88 and DOF reduction on different meshes. (a) mesh with 100 × 50. (b) mesh with 200 × 100. (c) mesh with 300 × 150.

considering only half the MBB beam, the external load, design domain and boundary conditions are shown in Fig. 11. This example aims to study the efficiency of DOF reduction, convergence acceleration and the whole acceleration algorithm.

5.1.1. DOF reduction

In this example, for the solution time of the finite element equations per iteration, the DOF reduction is compared with the efficient top88 code on different meshes. The parameters of the DOF reduction are set as follows: the error tolerance $tol = 0.1$ and displacement change threshold $\varepsilon = 0.001$. The volume fraction of the MBB beam is set to 0.5, and the design domain of the MBB beam is discretized into three different sizes: 100 × 50, 200 × 100 and 300 × 150 elements. Fig. 12 shows the solution time comparison between DOF reduction method and the top88 code on the three different meshes. As expected, due to the DOF reduction of the finite element equations, the solution

Table 6
Iteration number and compliance comparison between top88 and the two schemes on different meshes.

Case	Iteration number			Compliance		
	Top 88	Scheme 1	Scheme 2	Top 88	Scheme 1	Scheme 2
100 × 50	283	54	36	78.89	79.70	76.95
200 × 100	353	67	38	79.40	80.08	79.05
300 × 150	352	82	42	80.88	81.51	81.05

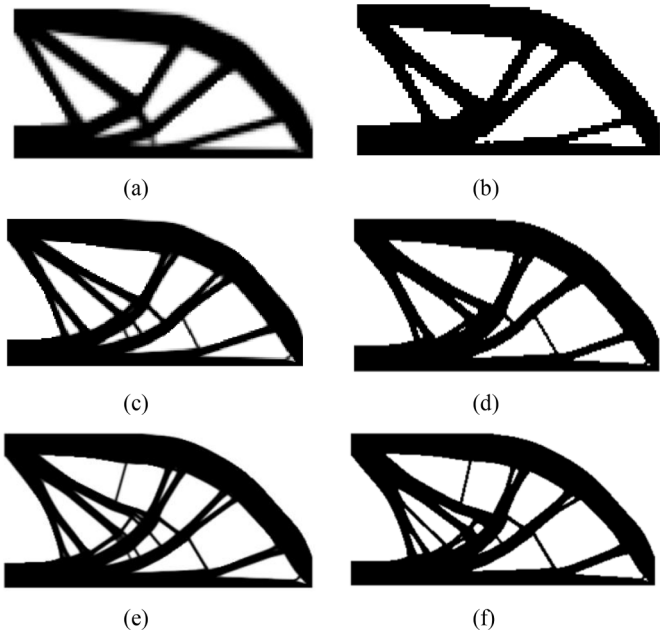

Fig. 14. The final optimized structures of top88 and the acceleration algorithm on different meshes. (a), (c) and (e) top88 with 100 × 50, 200 × 100 and 300 × 150 elements respectively. (b), (d) and (f) the acceleration algorithm with 100 × 50, 200 × 100 and 300 × 150 elements erespectively.

time of each iteration using DOF reduction method is less than that of the top88 code, which demonstrates the DOF reduction can effectively accelerate the solution time of the finite element equations. It can be also found from Fig. 12 that as the mesh scale increases, the solution time difference between the DOF reduction and top88 is larger, which implies the DOF reduction method has advantage in large-scale problems.

For the pre-processing for FEA, what is different between DOF reduction and top88 is that the DOF reduction requires some additional operations including calculating nodal densities, finding the "void" nodes, calculating displacement change, finding the small-change displacements and so on. For a fair comparison, we also test the total time for FEA (i.e., the sum of the solution time of FEA and the pre-processing time for FEA) and the result is shown in Fig. 13. It can be seen from Fig. 13 that the pre-processing of DOF reduction is not time-consuming compared to the pre-processing of top88. The efficiency of FEA is still improved considering the pre-processing time for FEA.

5.1.2. Convergence acceleration

The number of iterations is a key factor determining the total time of the topology optimization. This section compares the iteration number of top88 with that of the convergence acceleration methods under the same convergence criteria: the maximum difference of all element densities between two successive iterations is less than 0.01.

In Section 4.2, two schemes have been proposed to accelerate the convergence, one is reducing design variables, and the other is gray-scale suppression. For the sake of convenience, reducing design variables is called Scheme 1 and gray-scale suppression is called Scheme 2. The number of iterations and the compliance of the two schemes will be also compared to evaluate their respective efficiency and accuracy. The parameters required for Scheme 1 and Scheme 2 mentioned in algorithm 2 and algorithm 3 in Section 4.2 are set as follows: the objective change threshold $\delta = 0.005$ and $a = 0.6/obj\_ch$. The volume fraction is set to 0.5, and the design domain is discretized into three different mesh

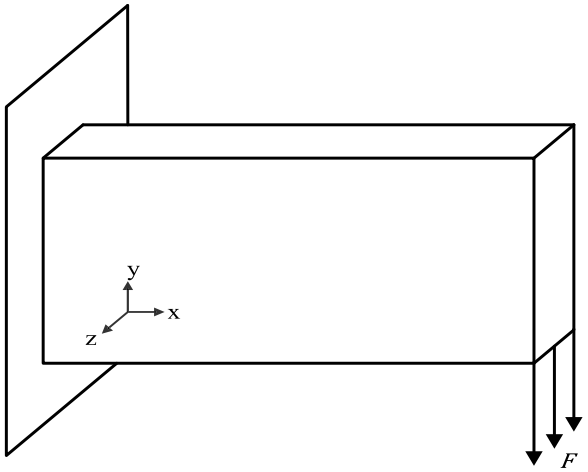

Fig. 15. The external load, design domain and boundary conditions of the 3D cantilever beam.

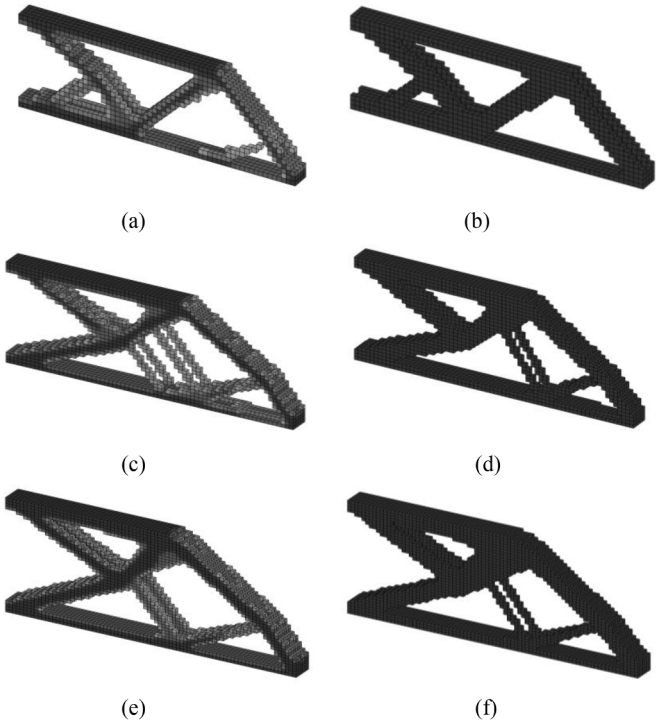

Fig. 16. The final optimized structures of top3d and the acceleration algorithm on different meshes. (a), (c) and (e) top3d with 60 × 20 × 4, 70 × 3 × 5 and 80 × 40 × 6 elements respectively. (b), (d) and (f) the acceleration algorithm with 60 × 20 × 4, 70 × 3 × 5 and 80 × 40 × 6 elements respectively.

Table 7
Efficiency comparison between top88 and the acceleration algorithm

Case	Compliance Top88 (acceleration)	Iteration number Top88 (acceleration)	Computational time (s) Top88 (acceleration)	$\text{Speedup}_{\frac{\text{time}_{\text{Top88}}}{\text{time}_{\text{acceleration}}}}$
100 × 50	78.89 (76.94)	283 (35)	63.8 (10.4)	6.1
200 × 100	79.40 (79.01)	353 (40)	215.6 (46.2)	4.7
300 × 150	80.88 (80.95)	361 (42)	1003.4 (74.2)	13.5

Table 8
Efficiency comparison between top 3d and the acceleration algorithm.

Case	Compliance Top3d (acceleration)	Iteration number Top3d (acceleration)	Computation time (s) Top3d (acceleration)	$\text{Speedup}\frac{\text{time}_{\text{Top3d}}}{\text{time}_{\text{acceleration}}}$
60 × 20 × 4	2417.67 (1718.86)	151 (46)	396.2 (76.6)	5.2
70 × 30 × 5	1390.77 (1086.64)	346 (46)	2308.7 (313.1)	7.4
80 × 40 × 6	1038.12 (864.14)	260 (47)	3403.4 (546.9)	6.2

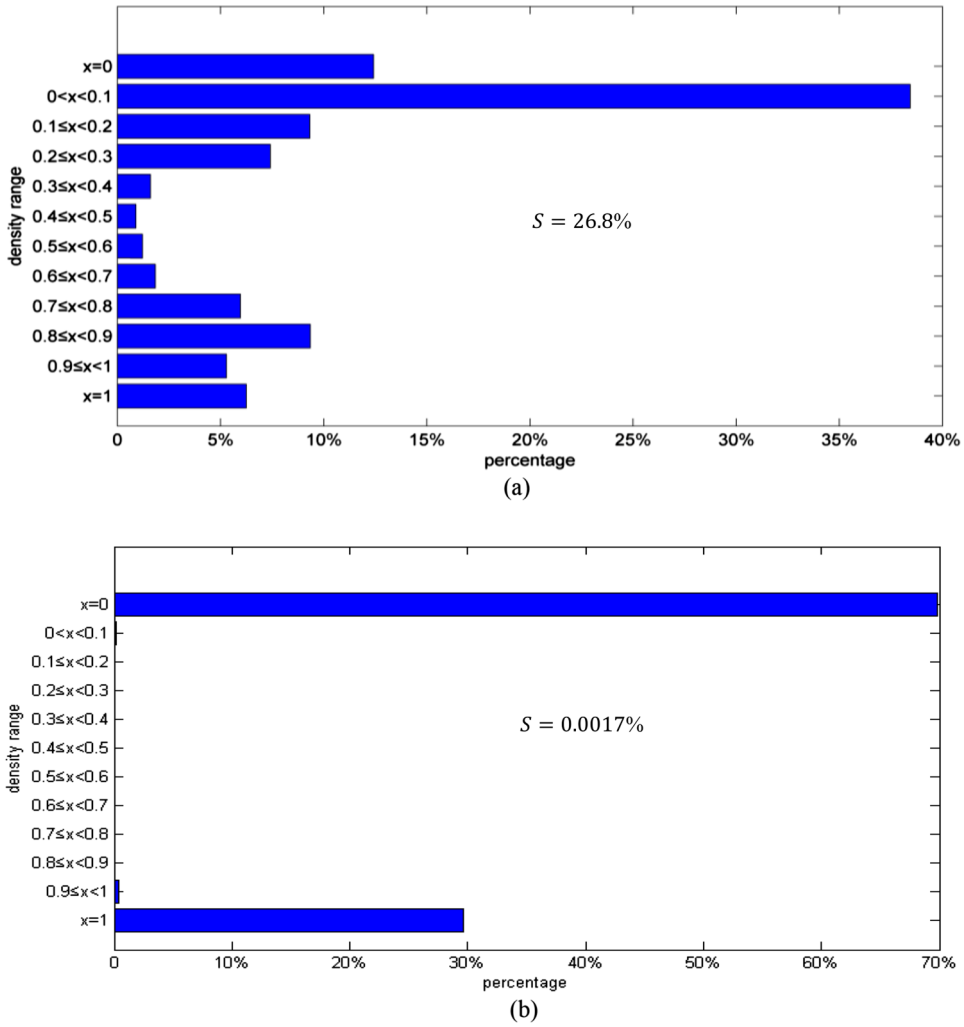

Fig. 17. The comparison of density distribution and the value of S between top3d and the acceleration algorithm. (a) top3d. (b) acceleration algorithm.

Table 9
The results of topology optimization with different δ

Parameter δ	0.001	0.005	0.01	0.05	0.1	0.5	1
Iteration number	47	45	41	36	33	29	19
Compliance	1715.50	1718.86	1715.79	1718.50	1720.50	1883.53	1999.35

sizes: 100 × 50, 200 × 100 and 300 × 150 elements.

Table 6 shows the comparison of iteration numbers and the compliance between top88 and the two schemes. From Table 6, it can be seen that the number of iterations required for convergence is greatly reduced for both Scheme 1 and Scheme 2, which demonstrates the reduction of design variables and the gray-scale suppression can greatly accelerate the convergence. It can be also found that the number of iterations of Scheme 2 is fewer than that of Scheme 1 for all the three different cases. The reason for the above is that some intermediate densities have not been reduced and are not very stable in the method of reducing design variables, which affect the convergence. By further studying the comparison of the compliance of top88 and the two schemes, we can find that the compliance of Scheme 1 is slightly larger than that of top88, exceeded percentage ranging from 0.78% to 1.03%, while the compliance of Scheme 2 is smaller than that of top88 in the first two cases, which means the optimized structure is more accurate since the polarization of density reduces the gray elements. It can be also found that the compliance of Scheme 2 is slightly larger than that of top88 in the third case, exceeded percentage is 0.21%. In practice, the error is very small that can be ignored.

In summary, both Scheme 1 and Scheme 2 can accelerate the convergence and obtain small-error compliance. Scheme 1 can reduce some

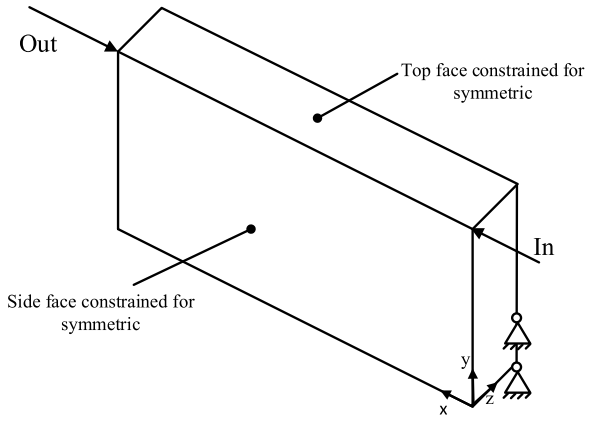

Fig. 18. The external load, design domain and boundary conditions of the 3D force inverter problem.

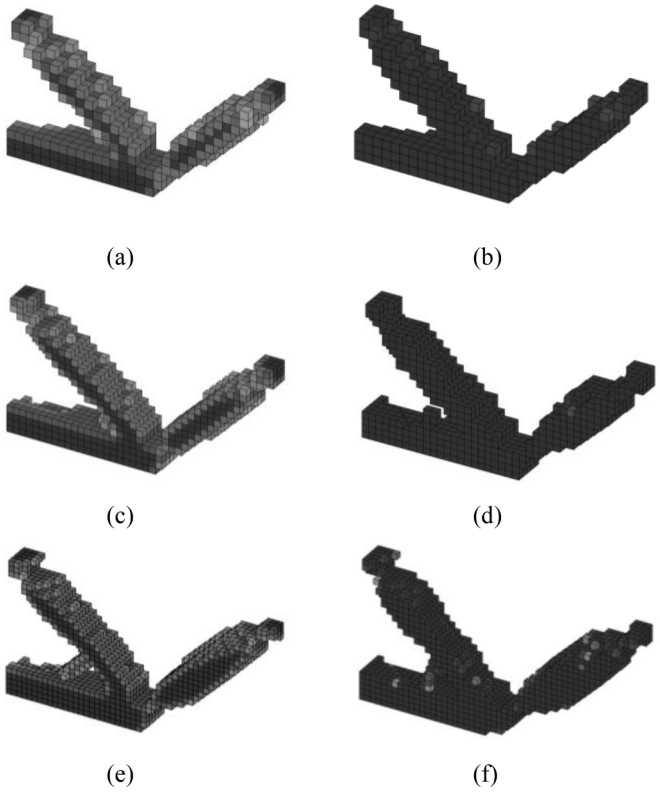

Fig. 19. The final optimized structures of topcm and the acceleration algorithm on different meshes. (a), (c) and (e) topcm with 30 × 15 × 4, 40 × 20 × 5 and 50 × 25 × 6 elements respectively. (b), (d) and (f) the acceleration algorithm with 30 × 15 × 4, 40 × 20 × 5 and 50 × 25 × 6 elements respectively.

design variables and therefore the time for updating the design variables is reduced. Scheme 2 can obtain more accurate compliance with fewer iterations compared to Scheme 1. In the subsequent numerical examples, Scheme 2 is adopted to test the efficiency of the acceleration algorithm.

5.1.3. Acceleration algorithm

In this section, we will study the efficiency of the acceleration algorithm which combines DOF reduction and convergence acceleration. The parameters required for the acceleration algorithm are the same as that in Section 5.1.1 and Section 5.1.2. The parameter P not mentioned above is set to 10. The design domain is discretized into 100 × 50, 200 × 100 and 300 × 150 elements. Fig. 14 shows the final optimized structures of top88 and the acceleration algorithm on the three different meshes. From Fig. 14, it can be seen that the final structures optimized by the acceleration algorithm are almost the same as that optimized by top88, which demonstrates the final structure optimized by the acceleration algorithm is credible. Due to the polarization of the intermediate densities, the structure optimized by the acceleration algorithm has almost no gray elements and the boundary looks clearer.

Table 7 presents the efficiency comparison between top88 and the acceleration algorithm. Due to the usage of DOF reduction and convergence acceleration, the total computational time of the acceleration algorithm is greatly reduced and the speedup value ranges from 4.7 to 13.5, which demonstrates the high efficiency of the acceleration algorithm. The compliance of the acceleration algorithm is smaller than that of top88 in the first two cases owing to the polarization of the gray elements. Though the compliance is slightly larger in the third case, the exceed percentage is only 0.09% that can be ignored. Therefore, the compliance obtained by the acceleration algorithm is accurate enough.

In conclusion, the acceleration algorithm greatly improved the computational efficiency with obtaining accurate structure and compliance, which will be further demonstrated by 3D numerical examples in the following sections.

5.2. 3D cantilever beam

The 3D cantilever beam [43] is utilized as an example to study the computational efficiency of the acceleration algorithm in 3D topology optimization. Fig. 15 presents the external load, design domain and boundary conditions of the 3D cantilever.

The classic top3d code [43] for the 3D topology optimization is used to compare with the acceleration algorithm. The parameters required for the acceleration algorithm in the 3D topology optimization are consistent with the 2D case mentioned above. The volume fraction is set to 0.3 and the design domain is discretized into three different meshes: 80 × 40 × 6, 70 × 30 × 5 and 60 × 20 × 4 linear cubic elements.

Fig. 16 shows the final optimized structures of top3d and the acceleration algorithm on the three different meshes and Table 8 presents the efficiency comparison between top3d and the acceleration algorithm. The computation time of the acceleration algorithm is still much less than that of top3d and the compliance is also smaller than that of top3d, which indicates that the acceleration algorithm is still highly efficient and can obtain accurate compliance for the 3D case. In MBB beam, the compliance of the acceleration algorithm is just slightly smaller than that of top88, while Table 8 shows that the compliance of the acceleration algorithm is much smaller than that of top3d. It can be roughly seen from Fig. 16 that there are many gray elements existing in the structure optimized by top3d, while the structure optimized by the acceleration algorithm is black and white. Owing to the existence of

Table 10
Efficiency comparison between topcm and the acceleration algorithm.

Case	Displacement Topcm (acceleration)	Iteration number Topcm (acceleration)	Computation time (s) Topcm (acceleration)	Speedup $\frac{time_{topcm}}{time_{acceleration}}$
30 × 15 × 4	-1.4905 (-1.6843)	305 (52)	112.8 (24.5)	4.6
40 × 20 × 5	-1.6466 (-1.7364)	270 (57)	536.7 (155.4)	3.4
50 × 25 × 6	-1.7495 (-1.7443)	>400 (39)	1977.5 (204.8)	9.8

Table 11
The iteration number for topcm in different cases.

Case	20 × 10 × 4	30 × 10 × 4	30 × 15 × 4	40 × 20 × 5	50 × 25 × 6	50 × 30 × 6
Iteration number for topcm	>400	>400	305	270	>400	>400

many gray elements, the material is not fully utilized, and therefore the compliance of top3d is much larger than the acceleration algorithm. A measure of discreteness [45] is adopted to quantitatively describe the degree of discreteness of an optimized structure, which is shown as follows:

$$S = \frac{\sum_{e=1}^{N} 4x_e(1-x_e)}{N} \times 100\% \tag{22}$$

Where S reflects the degree of discreteness. If there is no gray element in the optimized structure, i.e., all element densities are equal to 0 or 1, S is equal to 0. If the optimized structure is totally gray, i.e., all element densities are equal to 0.5, S is equal to 1. The smaller S is, the more an optimized structure tends to a discrete solution. The clear comparison of the density distribution and the value of S between top3d and the acceleration algorithm in the case $60 \times 20 \times 4$ is shown in Fig. 17. From Fig. 17(a), it can be seen that the proportion of the intermediate density is about 80% and $S = 26.8\%$, while in Fig. 17(b), $S = 0.0017\%$, i.e., there is almost no intermediate density in the structure optimized by the acceleration algorithm and the structure has converged to a discrete solution. We also calculate the value of S with the density distribution in Fig. 7, and the result is $S = 11.1\%$. Compared to $S = 26.8\%$ in the 3D case, $S = 11.1\%$ is relatively small in the 2D case. That is why there is a large compliance gap between top3d and the acceleration algorithm compared to the slight gap in the 2D case. From this perspective, the compliance of top3d is actually not accurate enough and the acceleration algorithm has a great advantage in obtaining accurate compliance and getting a discrete solution.

Section 3.2.2 has mentioned that the gray-scale suppression should not applied in the early stage of the topology optimization since the structure is not very stable at that stage. The formula Eq.(19) is applied to judge whether it is in the early stage and objective change threshold δ is the key parameter of the formula. Now we discuss how the choice of parameterδ affects the results of the topology optimization. The case $60 \times 20 \times 4$ is chosen as the study object. Table 9 shows the different iteration number and compliance with different δ. It can be roughly seen that as the parameter δ increases, the number of iterations decreases, but the compliance increases, which means the optimized structure may not be accurate. Consider that the compliance of top3d is 2417.67, which is still larger than 1999.35 when $\delta = 1$, so if we don't pursue very precise result, we can appropriately increase δ to obtain fewer iterations.

5.3. Compliant mechanism synthesis

This section uses a 3D force inverter problem [43] as an example to verify the efficiency of the acceleration algorithm. The code introduced in [43] is adopted to compare with the acceleration algorithm. To distinguish it from the previous top3d code, this code is named topcm here. The objective function of this problem is to maximize the negative horizontal output displacement with an input external load in the positive direction as shown in Fig. 18.

The volume fraction is set to 0.3 and the design domain is discretized into $30 \times 15 \times 4$, $40 \times 20 \times 5$ and $50 \times 25 \times 6$ linear cubic elements. Fig. 19 shows the final optimized structure of topcm and the acceleration algorithm on the three different meshes. From Fig. 19, it still can be seen that the structure optimized by the acceleration algorithm is credible with almost no gray elements and clear boundary.

Table 10 presents the efficiency comparison between topcm and the acceleration algorithm. Note that in the case $50 \times 25 \times 6$, the number of iterations of topcm is more than 400, i.e., after 400 iterations, topcm still does not satisfy the convergence criterion and the displacement and computational time in the Table 10 is the value at the 400-th iteration.

From Table 10, the conclusion can be drawn that for the compliant mechanism problem, the acceleration algorithm still have quite good performance, which obtains accurate results with greatly reduced time. In the case $50 \times 25 \times 6$, the number of iterations of topcm is over 400, which implies the topcm may converge slowly in dealing with the compliant mechanism problem. It can be verified from Table 11 that it is really hard for topcm to converge in different cases. For some problems that are difficult to converge, it is a great choice to use the acceleration algorithm.

6. Conclusion

This paper has proposed an acceleration algorithm for the topology optimization, which includes DOF reduction and convergence acceleration methods. DOF reduction accelerates the solution of the finite element equations by reducing some DOFs from the finite element equations. Convergence acceleration reduces the number of iterations by accelerating the polarization of the design variables. Three benchmark numerical examples: MBB beam, 3D cantilever beam and compliant mechanism have been tested to verify the acceleration algorithm. For each example, the detailed comparison of the efficiency and the optimized results between the acceleration algorithm and the traditional topology optimization are given in this paper. All the examples have demonstrated that the acceleration algorithm greatly improves the computational efficiency for the topology optimization and obtains accurate optimized results. In addition, the acceleration algorithm has the advantages to get better objective value and to handle the problems that are hard to converge. In the future, the acceleration algorithm is expected to combine with GPU parallel computing [48,49] to further improve the efficiency of the topology optimization and apply to other types of topology optimization methods such as isogeometric topology optimization and lattice topology optimization [50,51].

CRediT authorship contribution statement

Wei Zheng: Conceptualization, Methodology, Software, Writing - original draft. **Yingjun Wang:** Supervision, Writing - review & editing. **Yongfeng Zheng:** Writing - review & editing. **Daicong Da:** Methodology, Validation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been supported by National Natural Science Foundation of China (51705158), Guangdong Basic and Applied Basic Research Foundation (2019A1515011783), and the Fundamental Research Funds for the Central Universities (2018MS45). These supports are gratefully acknowledged.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.advengsoft.2020.102890.

Reference

[1] Bendsøe MP, Kikuchi N. Generating optimal topologies in structural design using a homogenization method. Comput Meth Appl Mech Eng 1988;71:197–224.
[2] Bendsøe MP, Sigmund O. Material interpolation schemes in topology optimization. Arch Appl Mech 1999;69:635–54.
[3] Stolpe M, Svanberg K. An alternative interpolation scheme for minimum compliance topology optimization. Struct Multidiscip Optim 2001;22:116–24.
[4] Xie YM, Steven GP. Evolutionary structural optimization for dynamic problems. Comput Struct 1996;58:1067–73.
[5] Xie YM, Steven GP. Basic evolutionary structural optimization. Evolutionary structural optimization: Springer; 1997. p. 12-29.
[6] Querin OM, Steven GP, Xie YM. Evolutionary structural optimisation (ESO) using a bidirectional algorithm. Eng Comput 1998.
[7] Zheng Y, Da D, Li H, Xiao M, Gao L. Robust topology optimization for multi-material structures under interval uncertainty. Appl Math Model 2020;78:627–47.
[8] Zheng Y, Wang Y, Lu X, Liao Z, Qu J. Evolutionary Topology Optimization for Mechanical Metamaterials with Auxetic Property. Int J Mech Sci 2020:105638.
[9] Wang M, Wang X, Guo D. A level set method for structural topology optimization. Comput Meth Appl Mech Eng 2003;192:227–46.
[10] Xia Q, Shi T, Liu S, Wang MY. A level set solution to the stress-based structural shape and topology optimization. Comput Struct 2012;90:55–64.
[11] Wang Y, Benson DJ. Isogeometric analysis for parameterized LSM-based structural topology optimization. Comput Mech 2016;57:19–35.
[12] Wang Y, Liao Z, Shi S, Wang Z, Poh LH. Data-Driven Structural Design Optimization for Petal-Shaped Auxetics Using Isogeometric Analysis. CMES-Comp Model Eng Sci 2020;122:433–58.
[13] Wang Y, Wang Z, Xia Z, Poh LH. Structural Design Optimization Using Isogeometric Analysis: A Comprehensive Review. CMES-Comp Model Eng Sci 2018;117:455–507.
[14] Bruggi Matteo, Duysinx Pierre. Topology optimization for minimum weight with compliance and stress constraints. Struct Multidiscip Optim 2012;46:369–84.
[15] Luo Z, Chen L, Yang J, Zhang Y, Abdel-Malek K. Compliant mechanism design using multi-objective topology optimization scheme of continuum structures. Struct Multidiscip Optim 2005;30:142–54.
[16] Li Q, Steven GP, Querin OM, Xie YM. Shape and topology design for heat conduction by Evolutionary Structural Optimization. Int J Heat Mass Transf 1999;42:3361–71.
[17] Xia Q, Shi T. A cascadic multilevel optimization algorithm for the design of composite structures with curvilinear fiber based on shepard interpolation. Compos Struct 2018;188:209–19.
[18] Yoon GH. Topology optimization for stationary fluid–structure interaction problems using a new monolithic formulation. Int J Numer Methods Eng 2010;82:591–616.
[19] Jenkins N, Maute K. An immersed boundary approach for shape and topology optimization of stationary fluid-structure interaction problems. Struct Multidiscip Optim 2016;54:1191–208.
[20] Huang X, Zhou SW, Xie YM, Li Q. Topology optimization of microstructures of cellular materials and composites for macrostructures. Comput Mater Sci 2013;67:397–407.
[21] Long K, Han D, Gu X. Concurrent topology optimization of composite macrostructure and microstructure constructed by constituent phases of distinct Poisson's ratios for maximum frequency. Comput Mater Sci 2017;129:194–201.
[22] Zhang Y, Gao L, Xiao M. Maximizing natural frequencies of inhomogeneous cellular structures by Kriging-assisted multiscale topology optimization. Comput Struct 2020;230:106197.
[23] Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O. Efficient topology optimization in MATLAB using 88 lines of code. Struct Multidiscip Optim 2011;43:1–16.
[24] Bates DM. Direct Methods for Sparse Linear Systems by Timothy A. Davis Int Stat Rev 2007;75:260–1.
[25] Davis TA.Direct methods for sparse linear systems: Siam; 2006.
[26] Saad Y.Iterative Methods for Sparse Linear Systems1996.
[27] Amir O, Stolpe M, Sigmund O. Efficient use of iterative solvers in nested topology optimization. Struct Multidiscip Optim 2009;42:55–72.
[28] Amir O, Bendsøe MP, Sigmund O. Approximate reanalysis in topology optimization. Int J Numer Methods Eng 2009;78:1474–91.
[29] Carlberg K, Farhat C. An adaptive POD-Krylov reduced-order model for structural optimization. 8th World Congress on Structural and Multidisciplinary Optimization. 2009. p. 1–5.
[30] Kirsch U. Reduced basis approximations of structural displacements for optimaldesign. AIAA J 1991;29:1751–8.
[31] Amir O. Revisiting approximate reanalysis in topology optimization: on the advantages of recycled preconditioning in a minimum weight procedure. Struct Multidiscip Optim 2014;51:41–57.
[32] Long K, Gu C, Wang X, Liu J, Du Y, Chen Z, et al. A novel minimum weight formulation of topology optimization implemented with reanalysis approach. Int J Numer Methods Eng 2019;120:567–79.
[33] Gogu C. Improving the efficiency of large scale topology optimization through on-the-fly reduced order model construction. Int J Numer Methods Eng 2015;101:281–304.
[34] Liao Z, Zhang Y, Wang Y, Li W. A triple acceleration method for topology optimization. Struct Multidiscip Optim 2019;60:727–44.
[35] Xia L, Breitkopf P. A reduced multiscale model for nonlinear structural topology optimization. Comput Meth Appl Mech Eng 2014;280:117–34.
[36] Choi Y, Oxberry G, White D, Kirchdoerfer T. Accelerating topology optimization using reduced order models. Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States); 2019.
[37] Guest JK, Smith Genut LC. Reducing dimensionality in topology optimization using adaptive design variable fields. Int J Numer Methods Eng 2010;81:1019–45.
[38] Kim SY, Kim IY, Mechefske CK. A new efficient convergence criterion for reducing computational expense in topology optimization: reducible design variable method. Int J Numer Methods Eng 2012;90:752–83.
[39] Zhao L, Chen Z, Qiu H, Liang G. A PDOC Method for Topology Optimization Design. Advanced Intelligent Computing Theories and Applications. 6th International Conference on Intelligent Computing, ICIC. 2010. August 18-21, 2010 Proceedings2010.
[40] Okamoto Y, Masuda H, Kanda Y, Hoshino R, Wakao S. Convergence Acceleration of Topology Optimization Based on Constrained Level Set Function Using Method of Moving Asymptotes in 3-D Nonlinear Magnetic Field System. IEEE Trans Magn 2017;53:1–4.
[41] Groenwold AA, Etman L. A simple heuristic for gray-scale suppression in optimality criterion-based topology optimization. Struct Multidiscip Optim 2009;39:217–25.
[42] Logan DL.A first course in the finite element method: Cengage Learning; 2011.
[43] Liu K, Tovar A. An efficient 3D topology optimization code written in Matlab. Struct Multidiscip Optim 2014;50:1175–96.
[44] Sigmund O. A 99 line topology optimization code written in Matlab. Struct Multidiscip Optim 2001;21:120–7.
[45] Sigmund O. Morphology-based black and white filters for topology optimization. Struct Multidiscip Optim 2007;33:401–24.
[46] Buhl T, Pedersen CB, Sigmund O. Stiffness design of geometrically nonlinear structures using topology optimization. Struct Multidiscip Optim 2000;19:93–104.
[47] Wang Y, Liao Z, Ye M, Zhang Y, Li W, Xia Z. An efficient isogeometric topology optimization using multilevel mesh, MGCG and local-update strategy. Adv Eng Softw 2020:139.
[48] Martínez-Frutos J, Martínez-Castejón PJ, Herrero-Pérez D. Efficient topology optimization using GPU computing with multilevel granularity. Adv Eng Softw 2017;106:47–62.
[49] Xia Z, Wang Y, Wang Q, Mei C. GPU parallel strategy for parameterized LSM-based topology optimization using isogeometric analysis. Struct Multidiscip Optim 2017;56:413–34.
[50] Wang Y, Arabnejad S, Tanzer M, Pasini D. Hip implant design with three-dimensional porous architecture of optimized graded density. J Mech Des 2018;140:111406.
[51] Wang Y, Xu H, Pasini D. Multiscale isogeometric topology optimization for lattice materials. Comput Meth Appl Mech Eng 2017;316:568–85.