



Concurrent topology optimization of multiscale composite structures in Matlab

Jie Gao^{1,2} · Zhen Luo² · Liang Xia¹ · Liang Gao¹

Received: 12 June 2018 / Revised: 15 May 2019 / Accepted: 3 June 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract

This paper presents the compact and efficient Matlab codes for the concurrent topology optimization of multiscale composite structures not only in 2D scenario but also considering 3D cases. A modified SIMP approach (Sigmund 2007) is employed to implement the concurrent topological design, with an energy-based homogenization method (EBHM) to evaluate the macroscopic effective properties of the microstructure. The 2D and 3D Matlab codes in the paper are developed, using the 88-line 2D SIMP code (Struct Multidisc Optim 43(1): 1–16, 2011) and the 169-line 3D topology optimization code (Struct Multidisc Optim 50(6): 1175–1196, 2014), respectively. This paper mainly contributes to the following four aspects: (1) the code architecture for the topology optimization of cellular composite structures (ConTop2D.m and ConTop3D.m), (2) the code to compute the 3D isoparametric element stiffness matrix (elementMatVec3D.m), (3) the EBHM to predict the macroscopic effective properties of 2D and 3D material microstructures (EBHM2D.m and EBHM3D.m), and (4) the code to calculate the sensitivities of the objective function with respect to the design variables at two scales. Several numerical examples are tested to demonstrate the effectiveness of the Matlab codes, which are attached in the Appendix, also offering an entry point for new comers in designing cellular composites using topology optimization.

Keywords Concurrent topology optimization · Multiscale composite structures · SIMP · Energy-based homogenization method · MATLAB codes

1 Introduction

In the last two decades, topology optimization has experienced the significant development with a wide range of applications from traditional mechanics to fluids, heat transfer, electromagnetic, and multiphysics (Bendsøe and Sigmund 2003). It can be regarded as a numerical iterative procedure to find the best material distribution in the pre-defined design domain. By far, many different topology optimization methods have been developed, like the homogenization method (Bendsøe and Kikuchi 1988), the Solid Isotropic Material with Penalization (SIMP) (Zhou and Rozvany 1991; Bendsøe

and Sigmund 1999), the Evolutionary Structural Optimization (ESO) (Xie and Steven 1993), the Level Set Method (LSM) (Sethian and Wiegmann 2000; Wang et al. 2003; Allaire et al. 2004), the phase field method (e.g., Takezawa et al. 2010), and the recently proposed moving morphable component (MMC) method (e.g., Guo et al. 2014).

An iterative approach (Theocaris and Stavroulaki 1999) was developed to study the problem of the optimal material design of structures, with a series of pre-defined microstructures. Later, Rodrigues et al. (2002) proposed a computational procedure for the topology optimization of both material layout and the local material properties. In recent years, the concurrent topology optimization for multiscale composite structure has become a hot topic in the field. Figure 1 displays several existing designs of multiscale composite structures, and the concurrent topology optimization for multiscale composite structures can be roughly classified into two types.

In the first case, only one kind of material microstructures is considered in the topology optimization, where the macrostructure is periodically configured by a series of microstructures but with the same topology. The earlier works intended to study the effect of the macro loads and boundary conditions

Responsible Editor: Junji Kato

✉ Liang Gao
gaoliang@hust.edu.cn

¹ The State Key Lab of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, 1037 Luoyu Road, Wuhan 430074, Hubei, China

² The School of Mechanical and Mechatronic Engineering, University of Technology Sydney, 15 Broadway, Ultimo, NSW 2007, Australia

Fig. 1 Several concurrent multiscale designs. **a** Design 1 (Kato et al. 2014). **b** Design 2 (Liu et al. 2008; Chen et al. 2017). **c** Design 3 (Rodrigues et al. 2002). **d** Design 4 (Xia and Breitkopf 2014, 2017). **e** Design 5 (Li et al. 2018b). **f** Design 6 (Gao et al. 2019b)

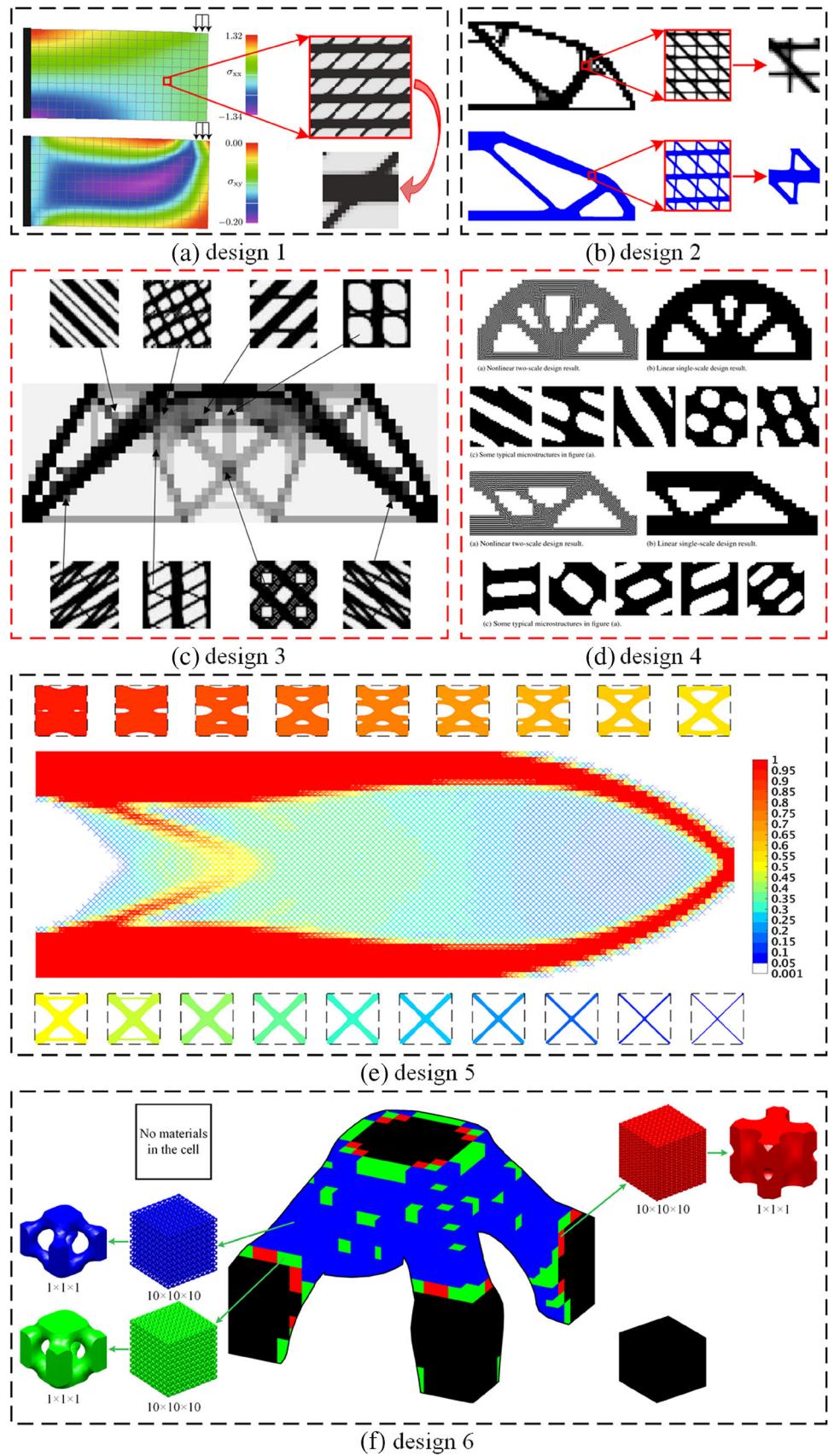
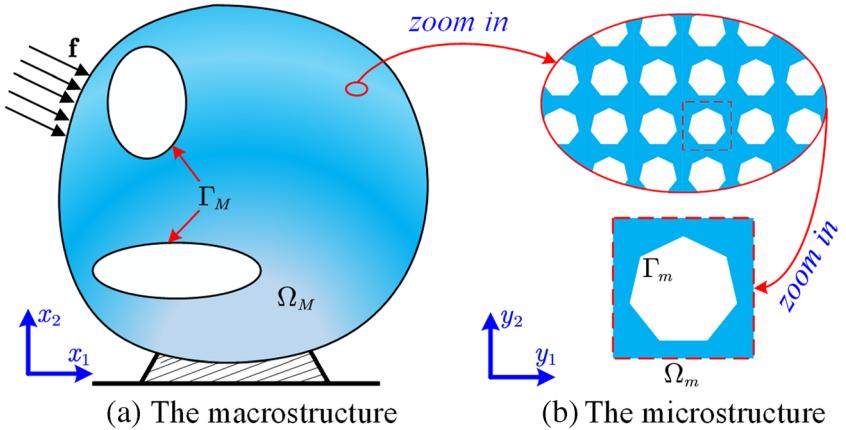


Fig. 2 A 2D two-scale structure.

a The macrostructure. **b** The microstructure



on the optimization of material microstructure, where the macrostructural topology keeps unchanged during the optimization (Huang et al. 2013; Kato et al. 2014), shown in Fig. 1a. Later, the topology at the macroscale was also optimized with the evolution of the topology of material microstructure (Liu et al. 2008; Yan et al. 2014; Wang et al. 2016, 2017b; Chen et al. 2017; Zheng et al. 2019), shown in Fig. 1b. This type of the concurrent topology optimization with a kind of material microstructures has the simple formulation, and no connectively issue is occurred in the optimization. Recently, this kind of designs has been applied to many problems, such as the multiphase (Da et al. 2017; Long et al. 2018), the dynamic (Niu et al. 2009; Zuo et al. 2013; Yan et al. 2015; Vicente et al. 2016), the thermal design (Yan et al. 2015; Xu et al. 2016), the robust (Guo et al. 2015; Zheng et al. 2019), the manufacturing (Yan et al. 2017), and the non-linear analysis (Kato et al. 2018).

The second type takes account of the multiscale composite structures with multiple kinds of material microstructures in the optimization. In Xia and Breitkopf (2014), a FE^2 non-linear multiscale analysis framework was developed for the concurrent topology optimization, and two designs are also shown in Fig. 1d. Since a large number of microstructures were simultaneously optimized and distributed in a point-wise manner, the computational cost was prohibitive. A decomposition method to reduce the number of designable microstructures and even the parallelization computation was employed to save the cost

(Sivapuram et al. 2016). Li et al. (2018b) developed a concurrent topology optimization method with multipatch microstructures by a parametric LSM (Wang and Wang 2006; Luo et al. 2007, 2008), and a design is displayed in Fig. 1e. Recently, Gao et al. (2019a, b) proposed a concurrent topology optimization formulation, where the macro and micro topologies, as well as the overall distribution of material microstructures using a density criterion to determine the total number of material microstructures to be designed, are all considered, and a simple concurrent design for 3D structures is given in Fig. 1f. Xu and Cheng (2018) also used the principal stress in the macrostructure to determine the total number of designable microstructures. Wang et al. (2018a) also studied the concurrent topology optimization for structures with non-uniform lattice microstructures using a porous anisotropic material with penalization model. Although the consideration of multiple kinds of material microstructures can improve the structural performance to a great extent, an important issue is introduced in the connectivity between different material microstructures (Wang et al. 2017a, 2018b; Li et al. 2018a, b; Du et al. 2018; Gao et al. 2019a). A wide range of applications of the concurrent topology optimization with multiple kinds of material microstructures has been also discussed in (Zhang and Sun 2006; Paulino et al. 2009; Alexandersen and Lazarov 2015; Li et al. 2016; Groen and Sigmund 2018; Chu et al. 2019; Zhang et al. 2019).

In the previous works, material effective properties are estimated using the numerical homogenization method (Guedes

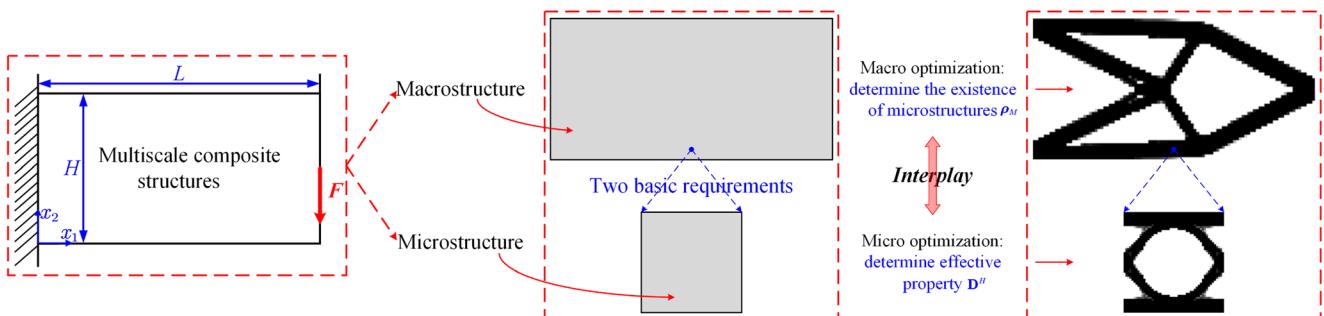
**Fig. 3** Concurrent topology optimization for multiscale composite structures

Table 1 Structural parameters in the 3D code

Array	Field	Explanation
Macro	Macro.length	The length of the macrostructure in x1-direction
	Macro.width	The width of the macrostructure in x2-direction
	Macro.height	The height of the macrostructure in x3-direction
	Macro.nelx	The number of macroelements in x1-direction
	Macro.nely	The number of macroelements in x2-direction
	Macro.nelz	The number of macroelements in x3-direction
	Macro.Vol	The maximal volume fraction of the macrostructure
	Macro.Elex	The length of the macroelement in x1-direction
	Macro.Eley	The width of the macroelement in x2-direction
	Macro.Elez	The height of the macroelement in x3-direction
	Macro.nele	The number of macroelements in the macrostructure
	Macro.ndof	The number of degrees of freedom in the macrostructure
	Micro.length	The length of the microstructure in y1-direction
	Micro.width	The width of the microstructure in y2-direction
	Micro.height	The height of the microstructure in y3-direction
Micro	Micro.nelx	The number of microelements in y1-direction
	Micro.nely	The number of microelements in y2-direction
	Micro.nelz	The number of microelements in y3-direction
	Micro.Vol	The maximal volume fraction of the microstructure
	Micro.Elex	The length of the microelement in y1-direction
	Micro.Eley	The width of the microelement in y2-direction
	Micro.Elez	The height of the microelement in y3-direction
	Micro.nele	The number of microelements in the microstructure
	Micro.ndof	The number of degrees of freedom in the microstructure

and Kikuchi 1990). The key concept is the prediction of the macroscopic effective properties of materials with the local heterogeneous behavior by the information of the microstructure, when the characteristic scales of inclusions (e.g., microstructures) are much smaller than that of the bulk sample (Hashin 1983). The homogenization has been extensively applied in many physical and engineering fields to evaluate the macroscopic effective properties of the continuous media (Xu et al. 2015a, b). As far as the numerical implementations, an energy-based homogenization method (EBHM) (Sigmund 1994; Xia and Breitkopf 2015) was established using the average stress theorems (Hashin 1983) and then combined with topology optimization to optimize microstructured materials (Zhang et al. 2007; Gao et al. 2018a, b, 2019c).

Since the 99-line Matlab code (Sigmund 2001), several educational articles with the compact Matlab codes and implementations have been published, which significantly contributed to the promotion of the topology optimization method in this field. For instance, Andreassen et al. (2011) developed a 88-line code as a complementary contribution to the 99-line code, and the 169-line Matlab code for 3D topology optimization using the SIMP method was given in Liu and Tovar (2014). Later, the Matlab implementations for the ESO-type method have been released by Huang and Xie

(2010). Recently, there have been Matlab codes for the level set-based methods, e.g., the discrete LSM code (Challis 2010) and the LSM with the reaction-diffusion equation (Otomori et al. 2014). Matlab codes for topology optimization were also provided in Suresh (2010) and Talischi et al. (2012a, b).

The main intention of the current work is to lower the barrier of the concurrent topology optimization by providing 2D and 3D Matlab codes to facilitate new comers to this field. It will also serve as an entry-level tutoring for researchers who want to familiarize with the research topics, including microstructured materials design and numerical implementations of the EBHM. The rest of this paper is organized as follows: a brief description for the homogenization is given in Sect. 3, and Sect. 4 presents the details of the concurrent topology optimization formulation and the sensitivity analysis. The Matlab implementations are provided in Sect. 5, several 2D and 3D numerical examples are demonstrated in Sect. 6, and the concluding remarks are followed in Sect. 6.

2 Homogenization

In the homogenization, two assumptions need to be satisfied in evaluating the macroscopic effective properties (Guedes

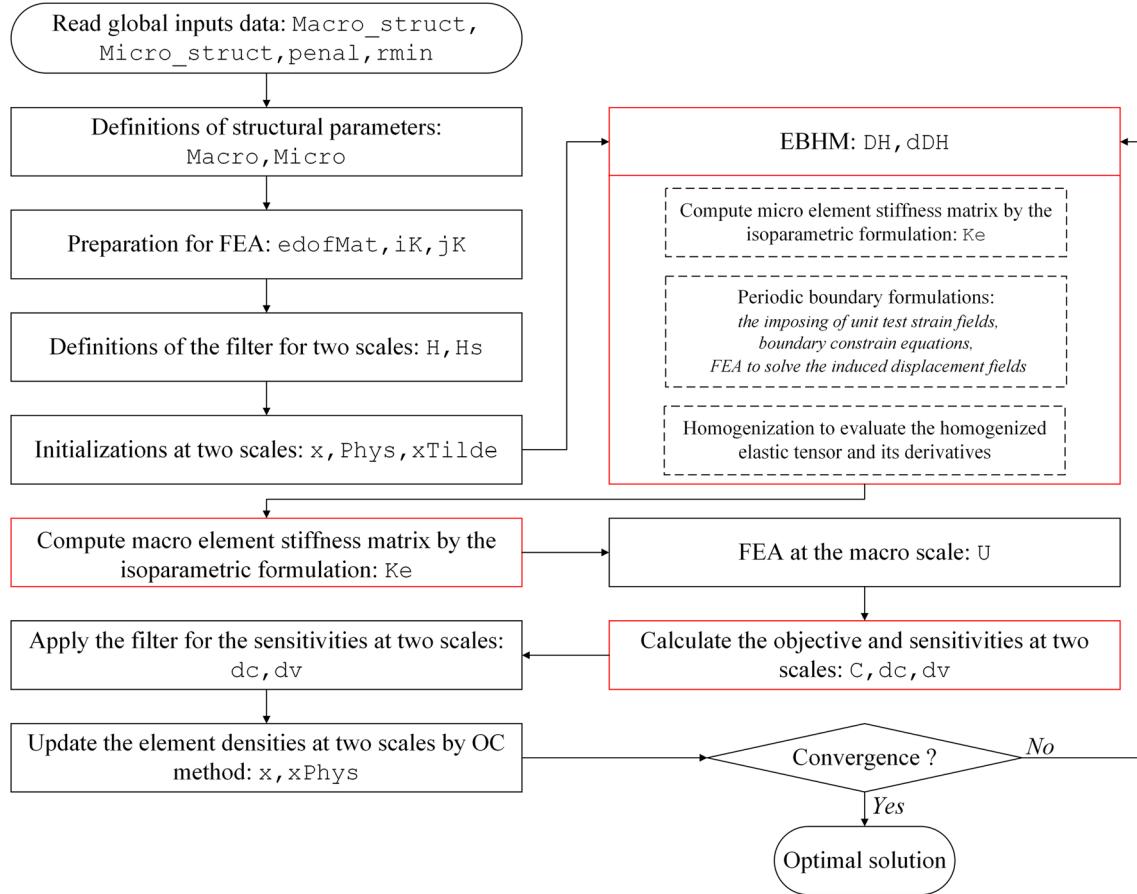


Fig. 4 The flowchart of 2D and 3D codes for the concurrent topology optimization

and Kikuchi 1990): (1) the macrosizes are much larger than that of the micro and (2) the microstructure is periodically distributed in the macrostructure. As shown in Fig. 2, a multiscale structure is provided, where the global coordinate system \mathbf{x} expresses the macroscale and the microscale is described in the local coordinate system \mathbf{y} .

Within the scope of the linear elasticity, the material elastic property $\mathbf{D}^e(\mathbf{x})$ is a \mathbf{y} -periodic function in the global coordinate system \mathbf{x} . ϵ is the aspect ratio between the scales of the microstructure and the macrostructure, which is far less than 1. The displacement field $\mathbf{u}^e(\mathbf{x})$ at the macrostructure can be characterized by the asymptotic expansion theory, given as follows:

$$\mathbf{u}^e(\mathbf{x}) = \mathbf{u}_0(\mathbf{x}, \mathbf{y}) + \epsilon \mathbf{u}_1(\mathbf{x}, \mathbf{y}) + \epsilon^2 \mathbf{u}_2(\mathbf{x}, \mathbf{y}) + \dots \quad (1)$$

For numerical simplicity without considering the dispersive behavior, only the first-order variation term with the small parameter expansion ϵ is considered, and the macroscopic stiffness tensor \mathbf{D}^H can be calculated by the following:

$$\mathbf{D}^H = \frac{1}{|\Omega_m|} \int_{\Omega_m} \mathbf{D} (\varepsilon(\mathbf{u}_m^0) - \varepsilon(\mathbf{u}_m)) (\varepsilon(\mathbf{u}_m^0) - \varepsilon(\mathbf{u}_m)) d\Omega_m \quad (2)$$

where $|\Omega_m|$ is the area (2D) or the volume (3D) of the microstructure, and \mathbf{D} is the locally varying elastic tensor. $\varepsilon(\mathbf{u}_m^0)$ is the linearly independent unit test strain field. $\varepsilon(\mathbf{u}_m)$ indicates the unknown strain field within the microstructure, which is solved by the following linear elasticity equilibrium equation with \mathbf{y} -periodic boundary conditions:

$$\int_{\Omega_m} \mathbf{D} \varepsilon(\mathbf{u}_m) \varepsilon(\boldsymbol{\nu}_m) d\Omega_m = \int_{\Omega_m} \mathbf{D} \varepsilon(\mathbf{u}_m^0) \varepsilon(\boldsymbol{\nu}_m) d\Omega_m, \quad (3)$$

$$\forall \boldsymbol{\nu}_m \in H_{per}(\Omega_m, \mathbb{R}^d)$$

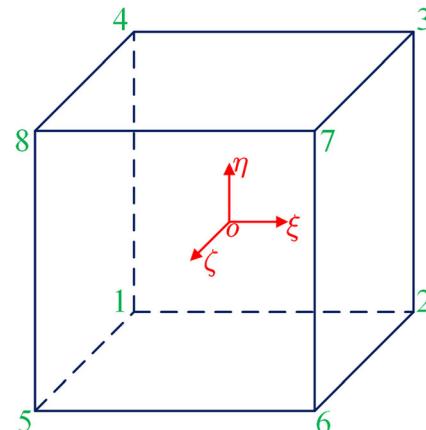


Fig. 5 A 3D iso-parametric element

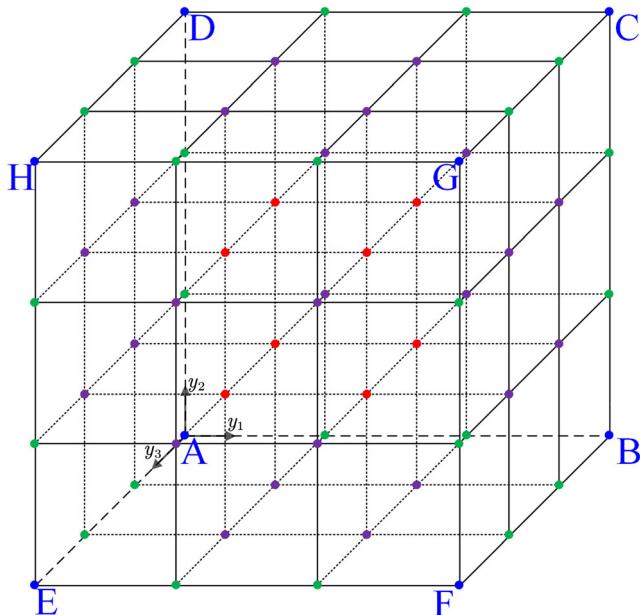


Fig. 6 A 3D material microstructure

where ν_m is the virtual displacement in the microstructure belonging to the admissible displacement space H_{per} with y-periodicity.

3 Concurrent topology optimization formulation

In the current paper, the first branch of the concurrent topology optimization with the consideration of only one kind of material microstructures is discussed to briefly show its key idea and also how to implement it using Matlab platform. The basic flowchart is presented in Fig. 3, where the macro and micro topologies are both taken account in the concurrent topology optimization design of multiscale composite structures to improve the structural performance. The former determines the macroscopic effective property, and the latter determines the spatial arrangement of the optimized microstructure in the macrostructure.

Fig. 7 Different types of nodes in the microstructure. **a** Nodes located at vertices. **b** Nodes located at edges. **c** Nodes located at surfaces. **d** Nodes located at the interior

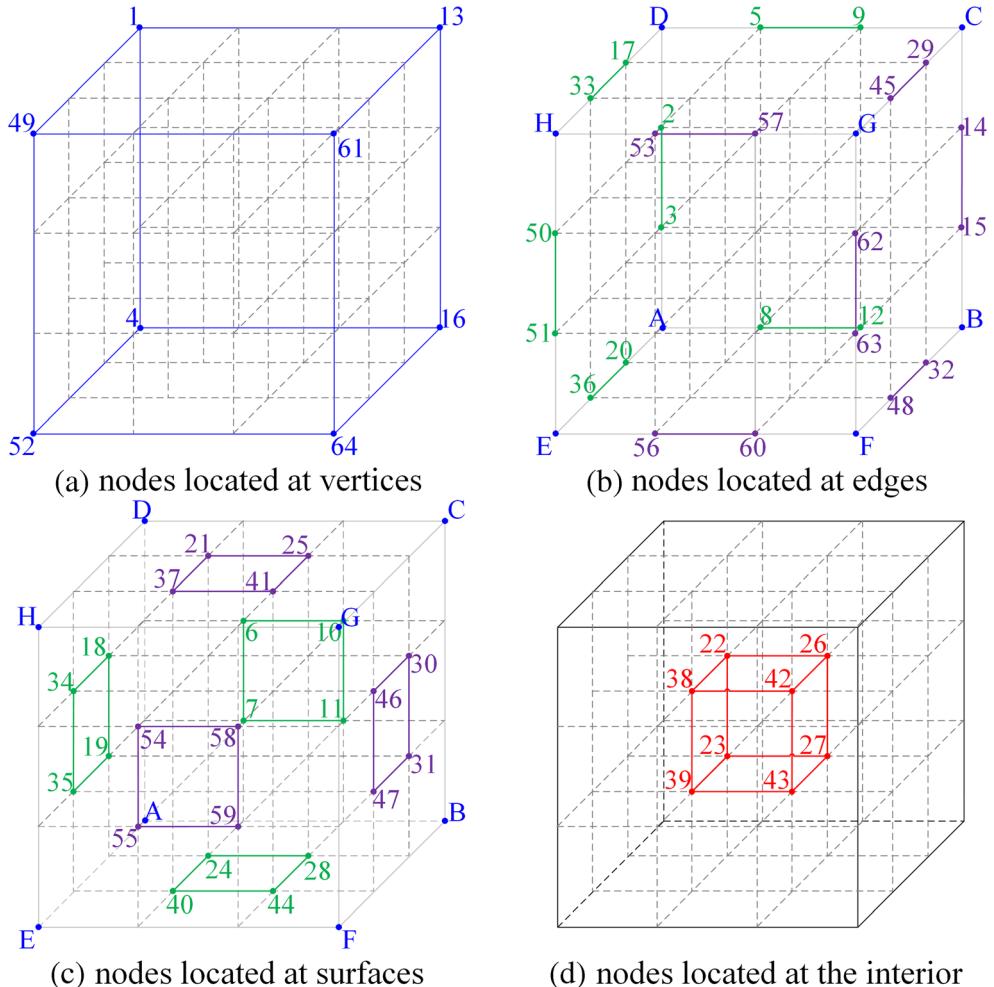


Table 2 Node sets in the microstructure

n1	4, 16, 13, 1, 52, 64, 61, 49 (vertices A, B, C, D, E, F, G, H)
n2	22, 23, 26, 27, 38, 39, 42, 43 (the interior of the PUC)
n3	20, 36, 17, 33, 8, 12, 5, 9, 2, 3, 50, 51, 6, 7, 10, 11, 18, 19, 34, 35, 24, 28, 40, 44 (edges: AE, DH, AB, DC, AD, EH; surfaces: ABCD, ADHE, ABFE)
n4	29, 45, 32, 48, 53, 57, 56, 60, 62, 63, 14, 15, 54, 55, 58, 59, 30, 31, 46, 47, 21, 25, 37, 41 (edges: CG, BF, HG, EF, FG, BC; surfaces: EFGH, BCGF, DCGH)

Currently, the corresponding mathematical model can be given as follows:

$$\left\{ \begin{array}{l} \text{Find: } \rho_M^i, \rho_m^j \quad (i = 1, 2, \dots, N_M; j = 1, 2, \dots, N_m) \\ \text{Min: } J(\rho_M, \rho_m) = \frac{1}{2} \int_{\Omega_M} \mathbf{D}_M(\rho_M, \rho_m) \varepsilon(\mathbf{u}_M) \varepsilon(\mathbf{u}_M) d\Omega_M \\ \text{S.t: } \begin{cases} a(\mathbf{u}_M, \boldsymbol{\nu}_M, \mathbf{D}_M) = l(\boldsymbol{\nu}_M), \quad \forall \boldsymbol{\nu}_M \in H_{per}(\Omega_M, \mathbb{R}^d) \\ a(\mathbf{u}_m, \boldsymbol{\nu}_m, \mathbf{D}_m) = l(\boldsymbol{\nu}_m), \quad \forall \boldsymbol{\nu}_m \in H_{per}(\Omega_m, \mathbb{R}^d) \\ G_M(\rho_M) = \int_{\Omega_M} v_0 \rho_M d\Omega_M - V_M \leq 0 \\ G_m(\rho_m) = \int_{\Omega_m} v_0 \rho_m d\Omega_m - V_m \leq 0 \\ 0 < \rho_M^{\min} \leq \rho_M^i \leq 1; 0 < \rho_m^{\min} \leq \rho_m^j \leq 1 \end{cases} \end{array} \right. \quad (4)$$

where J is the objective function, namely the structural mean compliance. The macrostructure Ω_M is discretized with N_M finite elements, and N_m finite elements are employed to discretize material microstructure Ω_m . The design variables at the macro- and microscales are symbolled by ρ_M and ρ_m , respectively. As we can see, the macro topology depended on ρ_M and the effective macroscopic property depended on ρ_m are both in the definition of J . G_M is the macro volume constraint, and V_M is the maximum material consumption of the macrostructure. G_m is the micro volume constraint, and V_m is the maximum volume fraction for material microstructure. v_0 is the volume fraction of solid element, equal to unit. ρ_M^{\min} and ρ_m^{\min} are the minimal values of the macro and micro variables, respectively, to avoid the numerical singularity in the optimization. \mathbf{u}_M is the macroscale displacement field, and $\boldsymbol{\nu}_M$ is the virtual displacement field of the macrostructure belonging to the space $H_{per}(\Omega_M, \mathbb{R}^d)$. a is the bilinear energy function. l is the linear load function. At the macroscale, the equilibrium state equation is established by the principle of virtual work, defined as follows:

$$\left\{ \begin{array}{l} a(\mathbf{u}_M, \boldsymbol{\nu}_M, \mathbf{D}_M) = \int_{\Omega_M} \mathbf{D}_M(\rho_M, \rho_m) \varepsilon(\mathbf{u}_M) \varepsilon(\boldsymbol{\nu}_M) d\Omega_M \\ l(\boldsymbol{\nu}_M) = \int_{\Omega_M} \mathbf{f} \boldsymbol{\nu}_M d\Omega_M + \int_{\Gamma_M} \mathbf{h} \boldsymbol{\nu}_M d\Gamma_M \\ a(\mathbf{u}_m, \boldsymbol{\nu}_m, \mathbf{D}_m) = \int_{\Omega_m} \mathbf{D}_m(\rho_m) \varepsilon(\mathbf{u}_m) \varepsilon(\boldsymbol{\nu}_m) d\Omega_m \\ l(\boldsymbol{\nu}_m) = \int_{\Omega_m} \mathbf{D}_m(\rho_m) \varepsilon(\mathbf{u}_m^0) \varepsilon(\boldsymbol{\nu}_m) d\Omega_m \end{array} \right. \quad (5)$$

where \mathbf{f} is the body force in the macrostructure and \mathbf{h} is the boundary traction on the Neumann boundary Γ_M of the macrostructure. \mathbf{D}_M and \mathbf{D}_m denote the stiffness tensors of the macrostructure and material microstructure, respectively, and which are defined by the material interpolation scheme in the modified SIMP approach (Sigmund 2007), given as follows:

$$\begin{cases} \mathbf{D}_M = [c + (\rho_M)^p(1-c)]\mathbf{D}^H \\ \mathbf{D}_m = [c + (\rho_m)^p(1-c)]\mathbf{D}_0 \end{cases} \quad (6)$$

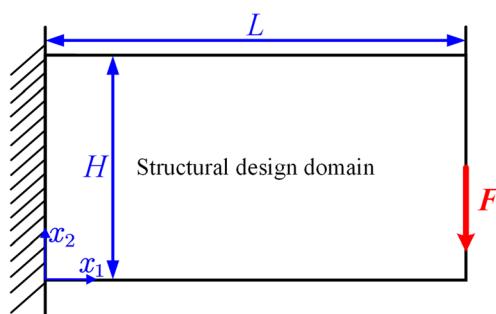
where \mathbf{D}_0 is the constitutive elastic tensor of the material, and $c = 1e^{-9}$ is a constant to avoid the singularity of the stiffness matrix. p is the penalty parameter. \mathbf{D}^H corresponds to the homogenized stiffness tensor, which is calculated based on the topology of the microstructure, as follows:

$$\mathbf{D}^H = \frac{1}{|\Omega_m|} \int_{\Omega_m} \mathbf{D}_m(\rho_m) (\varepsilon(\mathbf{u}_m^0) - \varepsilon(\mathbf{u}_m)) (\varepsilon(\mathbf{u}_m^0) - \varepsilon(\mathbf{u}_m)) d\Omega_m \quad (7)$$

The developed concurrent topology optimization formulation can be solved by many well-established gradient-based algorithms, like the optimality criteria (OC) method (Rozvany et al. 1995). Hence, the sensitivity for the objective and constraint functions with respect to the design variables is required. The first-order derivatives of the objective function and macro volume constraint with respect to the macro design variable are expressed as follows:

$$\begin{cases} \frac{\partial J}{\partial \rho_M} = -\frac{1}{2} \int_{\Omega_M} p(\rho_M)^{p-1}(1-c) \mathbf{D}^H(\rho_m) \varepsilon(\mathbf{u}_M) \varepsilon(\mathbf{u}_M) d\Omega_M \\ \frac{\partial G_M}{\partial \rho_M} = \int_{\Omega_M} v_0 d\Omega_M \end{cases} \quad (8)$$

The derivation of the first-order derivatives of the objective function and the micro volume constraint with respect to the micro design variables is slightly complex, and the details can refer to (Sigmund 1994; Xia and Breitkopf 2015; Gao et al. 2018a, b), and the final form is given as follows:

**Fig. 8** A cantilever beam

$$\begin{cases} \frac{\partial J}{\partial \rho_m} = -\frac{1}{2} \int_{\Omega_M} [c + (\rho_M)^p (1-c)] \frac{\partial \mathbf{D}^H(\rho_m)}{\partial \rho_m} \varepsilon(\mathbf{u}_M) \varepsilon(\mathbf{u}_M) d\Omega_M \\ \frac{\partial G_m}{\partial \rho_m} = \int_{\Omega_m} v_0 d\Omega_m \end{cases} \quad (9)$$

The first-order derivatives of the homogenized elastic tensor \mathbf{D}^H with respect to the micro element densities are given as follows:

$$\frac{\partial \mathbf{D}^H(\rho_m)}{\partial \rho_m} = \frac{1}{|\Omega_m|} \int_{\Omega_m} p(\rho_m)^{p-1} (1-c) \mathbf{D}_0 (\varepsilon(\mathbf{u}_m^0) - \varepsilon(\mathbf{u}_m)) (\varepsilon(\mathbf{u}_m^0) - \varepsilon(\mathbf{u}_m)) d\Omega_m \quad (10)$$

The detailed derivations for the OC method can refer to Bendsøe and Sigmund (2003). The updating factors for the advancing of the two-scale design variables can be obtained based on the Kuhn-Tucker conditions, given as follows:

$$\begin{cases} \rho_M^{i(K+1)} = (\Pi_M^{i(K)}) \rho_M^{i(K)} = \left(-\frac{\partial J}{\partial \rho_M}(\rho_M, \rho_m) / \max\left(\mu, \Lambda_M^{(K)} \frac{\partial G_m}{\partial \rho_M}\right) \right) \rho_M^{i(K)} \\ \rho_m^{j(K+1)} = (\Pi_m^{j(K)}) \rho_m^{j(K)} = \left(-\frac{\partial J}{\partial \rho_m}(\rho_M, \rho_m) / \max\left(\mu, \Lambda_m^{(K)} \frac{\partial G_m}{\partial \rho_m}\right) \right) \rho_m^{j(K)} \end{cases} \quad (11)$$

where K is the current iteration. $\Pi_M^{i(K)}$ and $\Pi_m^{j(K)}$ are the updating factors for the design variables $\rho_M^{i(K)}$ and $\rho_m^{j(K)}$, respectively, at the iteration K . μ is very small positive value to avoid

the zero in the denominator. $\Lambda_M^{(K)}$ and $\Lambda_m^{(K)}$ are the Lagrangian multipliers for the advancing of the two-scale design variables $\rho_M^{i(K)}$ and $\rho_m^{j(K)}$ at the current iteration K , respectively, which can be calculated by the bi-sectioning algorithm (Bendsøe and Sigmund 2003). The updating schemes of the macro and micro design variables are the same, and a unified form can refer to Bendsøe and Sigmund (2003). It is easily seen that the macro updating factor $\Pi_M^{i(K)}$ contains macro and micro design variables. Hence, the micro design variable ρ_m can influence the optimization of the macro topology, and the macro design variable ρ_M also affects the updating of the micro topology. That is, both the macro and micro updating factors $\Pi_M^{i(K)}$ and $\Pi_m^{j(K)}$ depend on the two-scale design variables ρ_M and ρ_m . Hence, the advancing of the macro and micro topologies is coupled in each iteration of the concurrent topology optimization for multiscale structures to improve the concerned performance.

4 Matlab implementations

In this section, we will comment on the major modifications based on the 88-line for 2D and 169-line for 3D Matlab codes to cater for the solution of the concurrent topology optimization problems. The main programs in 2D and 3D codes are called from the MATLAB prompt of the following lines:

```
1 ConTop2D(Macro_struct, Micro_struct, penal, rmin)
2 ConTop3D(Macro_struct, Micro_struct, penal, rmin)
```

where Macro_struct and Micro_struct are the related parameters of the macrostructure and microstructure, respectively. Structural parameters consist of the normal dimensions, the numbers of finite elements in the directions, and maximum volume fractions. In the main functions, two structure arrays (Macro and Micro) in Matlab codes are defined for the macrostructure and microstructure, respectively, which group the corresponding variable fields. An explanation in detail of arrays in 3D code is listed in Table 1, noticing that some parameters used in the 2D code have the same meanings. penal is the penalization parameter, and rmin is the filtering radius.

The provided 2D and 3D concurrent topology optimization Matlab codes inherit the basic framework of 2D and 3D SIMP codes, respectively. A basic flowchart for the concurrent topology optimization 2D and 3D codes is illustrated in Fig. 4. Compared with the 88-line code and 169-line code, there are some similar parts in the Matlab codes, such as the preparation for FEA, definitions of the Heaviside projection filter, initializations at two scales, and the OC algorithm.

The key differences between the concurrent and the conventional single-scale topology optimization are mainly involved into the following parts: (1) the effective macroscopic properties evaluated by the EBHM, including the 2D and 3D cases; (2) the iso-parametric formulation to calculate 2D and 3D element stiffness matrices; and (3) the sensitivity analysis of the objective function with the two-scale design variables, as shown in the red boxes of Fig. 4.

4.1 Subfunctions: elementMatVec2D and elementMatVec3D

Rectangular solid elements have been often adopted in most applications due to the easy formulation. However, the engineering products are featured with many complicated geometries, and it is difficult to mesh the structural design domain using only rectangular elements. The iso-parametric formulation (Cook et al. 1974) allows the quadrilateral and hexahedral elements to have non-rectangular shapes by introducing the reference coordinates to map the physical element into a reference element. In the physical

space, the element stiffness matrix is expressed as follows:

$$\mathbf{K}_e = \int_{\Omega_e} (\mathbf{B}_e)^T \mathbf{D} \mathbf{B}_e d\Omega_e \quad (12)$$

where Ω_e is the area (2D) or volume (3D) of the physical element. In the iso-parametric formulation, a reference coordinate system $(\xi - \eta - \zeta)$ is defined to describe the iso-parametric element, as shown in Fig. 5. After the transformation in two coordinates, the iso-parametric element stiffness matrix is evaluated in a new integration form, as follows:

$$\mathbf{K}_e = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 (\mathbf{B}_e)^T \mathbf{D} \mathbf{B}_e |\mathbf{J}| d\xi d\eta d\zeta \quad (13)$$

where \mathbf{J} is the Jacobian matrix.

We can see that the key factors of the integration are the strain-displacement matrix \mathbf{B}_e and Jacobian matrix \mathbf{J} . The strain-displacement relations of 3D element in the physical coordinate are stated in (14), where $u_{y_1} = \partial u / \partial y_1$ and the corresponding matrix is denoted by the symbol \mathbf{L} .

$$\begin{Bmatrix} \varepsilon_{y_1} \\ \varepsilon_{y_2} \\ \varepsilon_{y_3} \\ \gamma_{y_1 y_2} \\ \gamma_{y_2 y_3} \\ \gamma_{y_3 y_1} \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{Bmatrix} u_{y_1} \\ u_{y_2} \\ u_{y_3} \\ v_{y_1} \\ v_{y_2} \\ v_{y_3} \\ w_{y_1} \\ w_{y_2} \\ w_{y_3} \end{Bmatrix} = \mathbf{L} \begin{Bmatrix} u_{y_1} \\ u_{y_2} \\ u_{y_3} \\ v_{y_1} \\ v_{y_2} \\ v_{y_3} \\ w_{y_1} \\ w_{y_2} \\ w_{y_3} \end{Bmatrix} \quad (14)$$

The transformation between the derivatives of the displacement function with respect to the physical coordinates and the reference coordinates is shown in (15).

$$\begin{Bmatrix} \frac{\partial u}{\partial \xi} \\ \frac{\partial u}{\partial \eta} \\ \frac{\partial u}{\partial \zeta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial y_1}{\partial \xi} & \frac{\partial y_2}{\partial \xi} & \frac{\partial y_3}{\partial \xi} \\ \frac{\partial y_1}{\partial \eta} & \frac{\partial y_2}{\partial \eta} & \frac{\partial y_3}{\partial \eta} \\ \frac{\partial y_1}{\partial \zeta} & \frac{\partial y_2}{\partial \zeta} & \frac{\partial y_3}{\partial \zeta} \end{bmatrix} \begin{Bmatrix} \frac{\partial u}{\partial y_1} \\ \frac{\partial u}{\partial y_2} \\ \frac{\partial u}{\partial y_3} \end{Bmatrix} = \mathbf{J} \begin{Bmatrix} \frac{\partial u}{\partial y_1} \\ \frac{\partial u}{\partial y_2} \\ \frac{\partial u}{\partial y_3} \end{Bmatrix} \quad (15)$$

In the physical element, the position of a point is expressed by interpolating the shape functions on the nodal physical coordinates. Hence, the Jacobian matrix \mathbf{J} can be expressed by (16):

$$\mathbf{J} = \begin{bmatrix} N_{1,\xi} & N_{2,\xi} & N_{3,\xi} & N_{4,\xi} & N_{5,\xi} & N_{6,\xi} & N_{7,\xi} & N_{8,\xi} \\ N_{1,\eta} & N_{2,\eta} & N_{3,\eta} & N_{4,\eta} & N_{5,\eta} & N_{6,\eta} & N_{7,\eta} & N_{8,\eta} \\ N_{1,\zeta} & N_{2,\zeta} & N_{3,\zeta} & N_{4,\zeta} & N_{5,\zeta} & N_{6,\zeta} & N_{7,\zeta} & N_{8,\zeta} \end{bmatrix} \begin{bmatrix} y_1^1 & y_2^1 & y_3^1 \\ y_1^2 & y_2^2 & y_3^2 \\ y_1^3 & y_2^3 & y_3^3 \\ y_1^4 & y_2^4 & y_3^4 \\ y_1^5 & y_2^5 & y_3^5 \\ y_1^6 & y_2^6 & y_3^6 \\ y_1^7 & y_2^7 & y_3^7 \\ y_1^8 & y_2^8 & y_3^8 \end{bmatrix} \quad (16)$$

where (y_1^i, y_2^i, y_3^i) denotes the nodal coordinate in the physical element, and the shape functions are developed under the nodal ordering in the element displayed in Fig. 5, as expressed in (17).

$$N_i = \frac{1}{8} \begin{Bmatrix} (1-\xi)(1-\eta)(1-\zeta) \\ (1+\xi)(1-\eta)(1-\zeta) \\ (1+\xi)(1+\eta)(1-\zeta) \\ (1-\xi)(1+\eta)(1-\zeta) \\ (1-\xi)(1-\eta)(1+\zeta) \\ (1+\xi)(1-\eta)(1+\zeta) \\ (1+\xi)(1+\eta)(1+\zeta) \\ (1-\xi)(1+\eta)(1+\zeta) \end{Bmatrix} \quad (i = 1, 2, \dots, 8) \quad (17)$$

According to (15), the transformation for the derivatives of the three normal displacement fields with respect to two different coordinate systems can be expanded as a new form, given as follows:

$$\begin{Bmatrix} u_{y_1} \\ u_{y_2} \\ u_{y_3} \\ v_{y_1} \\ v_{y_2} \\ v_{y_3} \\ w_{y_1} \\ w_{y_2} \\ w_{y_3} \end{Bmatrix} = \begin{bmatrix} \mathbf{J}^{-1} & 0 & 0 \\ 0 & \mathbf{J}^{-1} & 0 \\ 0 & 0 & \mathbf{J}^{-1} \end{bmatrix} \begin{Bmatrix} u_\xi \\ u_\eta \\ u_\zeta \\ v_\xi \\ v_\eta \\ v_\zeta \\ w_\xi \\ w_\eta \\ w_\zeta \end{Bmatrix} = \mathbf{G} \begin{Bmatrix} u_\xi \\ u_\eta \\ u_\zeta \\ v_\xi \\ v_\eta \\ v_\zeta \\ w_\xi \\ w_\eta \\ w_\zeta \end{Bmatrix} \quad (18)$$

Meanwhile, the displacements of all points in 3D physical element are evaluated by the interpolation of the shape functions and nodal displacements. The corresponding derivatives of the displacements with respect to the reference coordinate system can be given as follows:

$$\begin{Bmatrix} u_\xi \\ u_\eta \\ u_\zeta \\ v_\xi \\ v_\eta \\ v_\zeta \\ w_\xi \\ w_\eta \\ w_\zeta \end{Bmatrix} = \begin{bmatrix} N_{1,\xi} & 0 & 0 & \dots & N_{8,\xi} & 0 & 0 \\ N_{1,\eta} & 0 & 0 & \dots & N_{8,\eta} & 0 & 0 \\ N_{1,\zeta} & 0 & 0 & \dots & N_{8,\zeta} & 0 & 0 \\ 0 & N_{1,\xi} & 0 & \dots & 0 & N_{8,\xi} & 0 \\ 0 & N_{1,\eta} & 0 & \dots & 0 & N_{8,\eta} & 0 \\ 0 & N_{1,\zeta} & 0 & \dots & 0 & N_{8,\zeta} & 0 \\ 0 & 0 & N_{1,\xi} & \dots & 0 & 0 & N_{8,\xi} \\ 0 & 0 & N_{1,\eta} & \dots & 0 & 0 & N_{8,\eta} \\ 0 & 0 & N_{1,\zeta} & \dots & 0 & 0 & N_{8,\zeta} \end{bmatrix} \quad \{d\} = dN\{d\} \quad (19)$$

where d is the nodal displacement matrix, and dN is the differentiation matrix of shape functions. Based on the element strain-displacement relation, the strain-displacement matrix \mathbf{B}_e is equal to the product of the above three rectangular matrices (\mathbf{L}, \mathbf{G} and dN), given as follows:

$$\mathbf{B}_e = \mathbf{L} \mathbf{G} dN \quad (20)$$

Finally, the Gauss numerical integration is employed to 2D and 3D iso-parametric element stiffness matrices, as given in elementMatVec2D.m and elementMatVec3D.m, respectively. The codes are called by the following lines:

```
1 Ke = elementMatVec2D(a, b, DH)
2 Ke = elementMatVec3D(a, b, c, DH)
```

where a , b , and c denote the half of the sizes of the element, respectively. DH is the elastic tensor. In the macrooptimization, DH denotes the homogenized elastic tensor (ConTop2D: line 48, ConTop3D: line 45). When calculating the element stiffness matrix of material microstructure (EBHM2D: line 6, EBHM3D: line

12), the elastic tensor matrix DH corresponds to the constitutive property \mathbf{D}_0 . It is a key ingredient of the concurrent topology optimization for multiscale composite structures. That is, the effective elastic tensor of the macrostructure is expressed as a function of the macro element density and the homogenized elastic

tensor. As given in the codes, the 2D and 3D element strain-displacement matrices \mathbf{B}_e are computed in line 13 (elementMatVec2D.m) and line 19 (elementMatVec3D.m), respectively, and the Matlab implementations are expressed as follows:

```

1  function Ke = elementMatVec2D(a, b, DH)
2  GaussNodes = [-1/sqrt(3); 1/sqrt(3)]; GaussWeigh = [1 1];
3  L = [1 0 0 0; 0 0 0 1; 0 1 1 0]; Ke = zeros(8,8);
4  for i = 1:length(GaussNodes)
5      for j = 1:length(GaussNodes)
6          GN_x = GaussNodes(i); GN_y = GaussNodes(j);
7          dN_x = 1/4*[-(1-GN_x) (1-GN_x) (1+GN_x) -(1+GN_x)];
8          dN_y = 1/4*[-(1-GN_y) -(1+GN_y) (1+GN_y) (1-GN_y)];
9          J = [dN_x; dN_y]*[ -a a a -a; -b -b b b]';
10         G = [inv(J) zeros(size(J)); zeros(size(J)) inv(J)];
11         dN(1,1:2:8) = dN_x; dN(2,1:2:8) = dN_y;
12         dN(3,2:2:8) = dN_x; dN(4,2:2:8) = dN_y;
13         Be = L*G*dN;
14         Ke = Ke + GaussWeigh(i)*GaussWeigh(j)*det(J)*Be'*DH*Be;
15     end
16 end
17 end

1  function Ke = elementMatVec3D(a, b, c, DH)
2  GN_x=[-1/sqrt(3),1/sqrt(3)]; GN_y=GN_x; GN_z=GN_x; GaussWeigh=[1,1];
3  Ke = zeros(24,24); L = zeros(6,9);
4  L(1,1) = 1; L(2,5) = 1; L(3,9) = 1;
5  L(4,2) = 1; L(4,4) = 1; L(5,6) = 1;
6  L(5,8) = 1; L(6,3) = 1; L(6,7) = 1;
7  for i=1:length(GN_x)
8      for j=1:length(GN_y)
9          for k=1:length(GN_z)
10              x = GN_x(i); y = GN_y(j); z = GN_z(k);
11              dNx = 1/8*[-(1-y)*(1-z) (1-y)*(1-z) (1+y)*(1-z) -(1+y)*(1-z)
12                  -(1-y)*(1+z) (1-y)*(1+z) (1+y)*(1+z) -(1+y)*(1+z)];
13              dNy = 1/8*[-(1-x)*(1-z) -(1+x)*(1-z) (1+x)*(1-z) (1-x)*(1-z)
14                  -(1-x)*(1+z) -(1+x)*(1+z) (1+x)*(1+z) (1-x)*(1+z)];
15              dNz = 1/8*[-(1-x)*(1-y) -(1+x)*(1-y) -(1+x)*(1+y) -(1-x)*(1+y)
16                  (1-x)*(1-y) (1+x)*(1-y) (1+x)*(1+y) (1-x)*(1+y)];
17              J = [dNx;dNy;dNz]*[ -a a a -a -a a a -a; -b -b b b -b
18                  b b; -c -c -c c c c c ]';
19              G = [inv(J) zeros(3) zeros(3);zeros(3) inv(J) zeros(3);zeros(3)
20                  zeros(3) inv(J)];
21              dN(1,1:3:24) = dNx; dN(2,1:3:24) = dNy; dN(3,1:3:24) = dNz;
22              dN(4,2:3:24) = dNx; dN(5,2:3:24) = dNy; dN(6,2:3:24) = dNz;
23              dN(7,3:3:24) = dNx; dN(8,3:3:24) = dNy; dN(9,3:3:24) = dNz;
24              Be = L*G*dN;
25              Ke = Ke +
26                  GaussWeigh(i)*GaussWeigh(j)*GaussWeigh(k)*det(J)*(Be'*DH*Be);
27      end
28  end
29 end

```

4.2 Subfunction: EBHM3D

The detailed information including the theoretical derivations and numerical implementations about the EBHM can refer to (Sigmund 1994; Xia and Breitkopf 2015). Xia and Breitkopf (2015) provides the descriptions for Matlab implementations of the 2D EBHM to optimize microstructured materials with the specific properties. However, the extension of the 2D to 3D is not straightforward. Gao et al. (2018a) recently provides a systematic derivation for the 3D EBHM, and the most important aspect is the development of the 3D periodic boundary

formulation within the material microstructure. Three basic elements are considered, including 3D periodic boundary conditions, 3D boundary constraint equations, and the reduced 3D linearly elastic equilibrium equation.

As shown in the attachment, an 84-line code for the 3D EBHM is divided into three components: the definitions of the material microstructure, the development of 3D periodic boundary formulation, and the homogenization to evaluate the macroscopic effective properties. The Matlab code is called from the prompt of the following line:

```
1 [CH, dCH] = EBHM3D(den, lx, ly, lz, E0, Emin, nu, penal)
```

where den is the element density matrix of the microstructure. A 3D microstructure (A–F) is assumed to be rectangular and discretized into 27 eight-node hexahedral elements, as shown in Fig. 6. lx, ly, and lz are the sizes of the material microstructure in three normal directions, respectively. The fifth and sixth arguments E0 and Emin are the Young's modulus for the material and void material, and nu is the Poisson's ratio. The last input penal is the penalization parameter.

parameters for the microstructure are grouped in the array Micro. Meanwhile, the subfunction elementMatVec3D is called in line 12 to calculate the 3D iso-parametric element stiffness matrix. After that, the 3D element index matrix edofMat is defined in lines 13 to 17 for the latter assembly of the global stiffness matrix. The detailed description for the index matrix is referred to Liu and Tovar (2014). The Matlab implementation is expressed as follows:

The definitions of the microstructure (lines 2 to 17) Similar to the definitions of the parameters of the macrostructure, the

```
2 % the initial definitions of the PUC
3 D0 = E0/(1+nu)/(1-2*nu)*...
4     [ 1-nu   nu   nu   0       0           0   ;
5      nu 1-nu   nu   0       0           0   ;
6      nu   nu 1-nu   0       0           0   ;
7      0     0   0   (1-2*nu)/2   0           0   ;
8      0     0   0       0   (1-2*nu)/2   0           ;
9      0     0   0       0           0   (1-2*nu)/2];
10 [nely, nelx, nelz] = size(den); nele = nelx*nely*nelz;
11 dx = lx/nelx; dy = ly/nely; dz = lz/nelz;
12 Ke = elementMatVec3D(dx/2, dy/2, dz/2, D0);
13 Num_node = (1+nely)*(1+nelx)*(1+nelz);
14 nodenrs = reshape(1:Num_node,1+nely,1+nelx,1+nelz);
15 edofVec = reshape(3*nodenrs(1:end-1,1:end-1,1:end-1)+1,nelx*nely*nelz,1);
16 edofMat = repmat(edofVec,1,24)+repmat([0 1 2 3*nely+[3 4 5 0 1 2] -3 -2 -1 ...
17 3*(nelx+1)*(nely+1)+[0 1 2 3*nely+[3 4 5 0 1 2] -3 -2 -1]], nele, 1);
```

3D periodic boundary formulation (lines 18 to 72) 3D periodic boundary formulation is developed in detail (Gao et al. 2018a) considering the 3D periodic boundary conditions, 3D boundary constraint equations, and the reduced 3D linearly elastic equilibrium equation. The Matlab code for 3D periodic boundary formulation is divided into the corresponding

subcomponents: the classifications of nodes in material microstructure to develop 3D boundary constraint equations (lines 18 to 42), the imposing of six independent unit test strains (lines 43 to 53), 3D boundary constraint equations (lines 54 to 63), and the reduced equilibrium equation to compute the induced displacement field (lines 64 to 72).

```

18 % 3D periodic boundary formulation
19 % the nodes classification
20 n1 = [nodenrs(end, [1 end], 1) nodenrs(1, [end 1], 1) nodenrs(end, [1 end], end)
21 nodenrs(1, [end 1], end)];
22 d1 = reshape([3*n1-2; 3*n1-1; 3*n1],3*numel(n1),1);
23 n3 =
24 [reshape(squeeze(nodenrs(end,1,2:end-1)),1,numel(squeeze(nodenrs(end,1,2:end
25 -1))))... % AE
26 reshape(squeeze(nodenrs(1, 1, 2:end-1)),1,numel(squeeze(nodenrs(1, 1,
27 2:end-1))))... % DH
28 reshape(squeeze(nodenrs(end,2:end-1,1)),1,numel(squeeze(nodenrs(end,2:end-1,
29 1))))... % AB
30 reshape(squeeze(nodenrs(1, 2:end-1, 1)),1,numel(squeeze(nodenrs(1,
31 2:end-1, 1))))... % DC
32 reshape(squeeze(nodenrs(2:end-1, 1, 1)),1,numel(squeeze(nodenrs(2:end-1,
33 1, 1))))... % AD
34 reshape(squeeze(nodenrs(2:end-1,1,end)),1,numel(squeeze(nodenrs(2:end-1,1,en
35 d)))... % EH
36 reshape(squeeze(nodenrs(2:end-1, 2:end-1,
37 1)),1,numel(squeeze(nodenrs(2:end-1, 2:end-1, 1))))... % ABCD
38 reshape(squeeze(nodenrs(2:end-1, 1,
39 2:end-1)),1,numel(squeeze(nodenrs(2:end-1, 1, 2:end-1))))... % ADHE
40 reshape(squeeze(nodenrs(end,2:end-1,2:end-1)),1,numel(squeeze(nodenrs(end,2:
41 end-1,2:end-1))))]; % ABFE
42 d3 = reshape([3*n3-2; 3*n3-1; 3*n3],3*numel(n3),1);
43 n4 = [reshape(squeeze(nodenrs(1, end, 2:end-1)),1,numel(squeeze(nodenrs(1, end,
44 2:end-1))))... % CG
45 reshape(squeeze(nodenrs(end,end,2:end-1)),1,numel(squeeze(nodenrs(end,end,2:
46 end-1))))... % BF
47 reshape(squeeze(nodenrs(1, 2:end-1, end)),1,numel(squeeze(nodenrs(1,
48 2:end-1, end))))... % HG
49 reshape(squeeze(nodenrs(end,2:end-1,end)),1,numel(squeeze(nodenrs(end,2:end-
50 1,end)))... % EF
51 reshape(squeeze(nodenrs(2:end-1,end,end)),1,numel(squeeze(nodenrs(2:end-1,en
52 d,end)))... % FG
53 reshape(squeeze(nodenrs(2:end-1, end,
54 1)),1,numel(squeeze(nodenrs(2:end-1, end, 1))))... % BC
55 reshape(squeeze(nodenrs(2:end-1,2:end-1,end)),1,numel(squeeze(nodenrs(2:end-
56 1,2:end-1,end)))... % EFGH
57 reshape(squeeze(nodenrs(2:end-1,end,2:end-1)),1,numel(squeeze(nodenrs(2:end-
58 1,end,2:end-1)))... % BCGF
59 reshape(squeeze(nodenrs(1, 2:end-1, 2:end-1)),1,numel(squeeze(nodenrs(1,
60 2:end-1, 2:end-1))))]; % DCGH
61 d4 = reshape([3*n4-2; 3*n4-1; 3*n4],3*numel(n4),1);
62 n2 = setdiff(nodenrs(:,[n1(:);n3(:);n4(:)])); d2 = reshape([3*n2-2; 3*n2-1;
63 3*n2],3*numel(n2),1);

```

As discussed in Xia and Breitkopf (2015) and Gao et al. (2018a), the 3D periodic boundary conditions are imposed on the opposite boundaries by constraining the displacements of the pairs of nodes located at the surfaces, edges, and vertices in the microstructure. For example, a microstructure shown in Fig. 6 is discretized into $3 \times 3 \times 3$ elements for the main intention of a simple illustration. The nodes in the microstructure are identified with a number ordered by column-wise up-to-bottom, left-to-right, and back-to-front. As displayed in Fig. 7, the microstructure has four types of the nodes subject to their corresponding locations (vertices, edges, surfaces, the interior). The nodes in the microstructure will be classified into four sets, listed in Table 2 and denoted by **n1** (lines 20 to 21), **n2** (line 42), **n3** (lines 22 to 31), and **n4** (lines 32 to 41), respectively. Node sets 1 (**n1**) and 2 (**n2**) consist of the nodes located

at the vertices and interior, respectively, and which are plotted by different colors (**n1**: blue and **n2**: red). The nodes plotted by the green and the purple in Fig. 7b, c corresponds to the node set 3 (**n3**) and node set 4 (**n4**), respectively, and which are located at the respective edges and surfaces in the material microstructure.

The six linearly independent test strains are defined in lines 43 to 53, and the initial test strains are directly imposed on the structural vertices (from A to H). The vertex A is fixed to avoid rigid body motion. The displacements of other vertices (B-H: lines 45 to 53) are computed according to the 3D boundary constraint equations. The displacement relationships of the nodes located at the opposite boundaries including the edges and surfaces in **n3** and **n4** sets are also computed by the corresponding 3D constraint equations (lines 54 to 63).

```

43 % the imposing of six linearly independent unit test strains
44 e = eye(6); ufixed = zeros(24,6);
45 vert_cor = [0, 1x, 1x, 0, 0, 1x, 1x, 0;
46           0, 0, ly, ly, 0, 0, ly, ly;
47           0, 0, 0, 0, lz, lz, lz, lz];
48 for i = 1:6
49     epsilon = [ e(i,1), e(i,4)/2, e(i,6)/2;
50                 e(i,4)/2, e(i,2), e(i,5)/2;
51                 e(i,6)/2, e(i,5)/2, e(i,3)];
52     ufixed(:,i) = reshape(epsilon*vert_cor,24,1);
53 end
54 % 3D boundary constraint equations
55 wfixed = [ repmat(ufixed(
56     7:9,:)),numel(squeeze(nodenrs(end,1,2:end-1))),1]; % C
56     repmat(ufixed( 4:6,:)-ufixed(10:12,:),numel(squeeze(nodenrs(1, 1,
2:end-1))),1); % B-D
57
      repmat(ufixed(22:24,:),numel(squeeze(nodenrs(end,2:end-1,1))),1);
      % H
58     repmat(ufixed(13:15,:)-ufixed(10:12,:),numel(squeeze(nodenrs(1,
2:end-1, 1))),1); % E-D
59     repmat(ufixed(16:18,:),numel(squeeze(nodenrs(2:end-1, 1,
1))),1); % F
60     repmat(ufixed(
4:6,:)-ufixed(13:15,:),numel(squeeze(nodenrs(2:end-1,1,end))),1); % B-E
61     repmat(ufixed(13:15,:),numel(squeeze(nodenrs(2:end-1, 2:end-1,
1))),1); % E
62     repmat(ufixed( 4:6,:),numel(squeeze(nodenrs(2:end-1, 1,
2:end-1))),1); % B
63
      repmat(ufixed(10:12,:),numel(squeeze(nodenrs(end,2:end-1,2:end-1))),1)];
      % D

```

Based on the developed boundary constraint equations and nodes classifications, the reduced linearly elastic equilibrium equation is derived (lines 64 to 72). It is noted that the global stiffness matrix of the microstructure is assembled using the

triplet to reduce the computational cost. iK and jK correspond to the row and column indices of the non-zero entries for the global stiffness matrix, which are created from edofMat and the spatial dimensions (lines 65 to 66). Meanwhile, the

element densities should be interpolated into the corresponding element stiffness matrix based on (8) (line 67). Finally, the unknown displacement fields induced by the imposing of initial test strain fields are evaluated in lines 70 to 72. The re-

duced linearly elastic equilibrium equation of the 3D microstructure provided in (Gao et al. 2018a) is sparse and consistent with the 2D (Xia and Breitkopf 2015) to make the 3D EBHM code more compact.

```

64 % the reduced elastic equilibrium equation to compute the induced displacement
  field
65 iK = reshape(kron(edofMat,ones(24,1))',24*24*nele,1);
66 jK = reshape(kron(edofMat,ones(1,24))',24*24*nele,1);
67 sK = reshape(Ke(:)*(Emin+den(:)).^penal*(1-Emin)),24*24*nele,1);
68 K = sparse(iK(:), jK(:), sK(:)); K = (K+K')/2;
69 Kr = [K(d2,d2), K(d2,d3)+K(d2,d4);
         K(d3,d2)+K(d4,d2), K(d3,d3)+K(d4,d3)+K(d3,d4)+K(d4,d4)];
70 U(d1,:) = ufixed;
71 U([d2;d3],:) =
    Kr\([-[K(d2,d1);K(d3,d1)+K(d4,d1)]*ufixed-[K(d2,d4);K(d3,d4)+K(d4,d4)]*wfix
    ed);
72 U(d4,:) = U(d3,:)+wfixed;

```

Homogenization to evaluate the effective material property (lines to 73 to 84): After obtaining the superimposed displacement field induced by the initial unit test strains, we evaluate the elementary mutual energies (line 79) of all finite elements. The summation of elementary mutual energies in the microstructure is calculated by a loop (lines 76 to 83), which corresponds to the homogenized elastic ten-

sor CH. Meanwhile, we compute the first-order derivatives of the homogenized elastic tensor with respect to the microelement densities in line 81 in order to compute the first-order derivatives of the objective with respect to the microelement densities. Finally, the numerical result for the evaluated elastic tensor is presented in line 84.

```

73 % homogenization to evaluate macroscopic effective properties
74 qe = cell(6,6); DH = zeros(6,6); dDH = cell(6,6);
75 cellVolume = lx*ly*lz;
76 for i = 1:6
77     for j = 1:6
78         U1 = U(:,i); U2 = U(:,j);
79         qe{i,j} =
            reshape(sum((U1(edofMat)*Ke).*U2(edofMat),2),nely,nelx,nelz);
80         DH(i,j) =
            1/cellVolume*sum(sum((Emin+den.^penal*(1-Emin)).*qe{i,j})));
81         dDH{i,j} = 1/cellVolume*(penal*(1-Emin)*den.^ (penal-1).*qe{i,j});
82     end
83 end
84 disp('--- Homogenized elasticity tensor ---'); disp(DH)

```

4.3 Sensitivity analysis at two scales

The sensitivities of the structural mean compliance at two scales are provided in (8), (9), and (10). We can see that the macro derivatives of the objective function consist of the micro element densities, and the derivatives of the structural mean compliance with respect to the micro design variables also include the macro element densities. Hence, the multiscale between the macro and micro designs is occurred within the optimization, which shows

the difference of the concurrent topology optimization compared to the single-scale problems. The details of the Matlab implementations are provided in the following section.

The codes in 2D and 3D for the two-scale sensitivities are shown in lines 52 to 66 (ConTop2D) and lines 49 to 66 (ConTop3D), respectively. It can be easily seen that the first-order sensitivities for the objective function and macro volume constraint with respect to the macro element densities (lines 53 to 56 in ConTop2D and lines 50 to 53 in ConTop3D) are analogous

to that of the single-scale problems (lines 57 to 61 in the 88-line 2D code and lines 74 to 77 in the 169-line 3D code). However, the key difference is how to compute the element stiffness matrices. It should be noted that the macrostructure is assumed to be periodically configured by the identical material microstructures in the concurrent topology optimization formulation for the multiscale composite structure. Hence, the effective elastic tensor within the evaluation of the macro stiffness matrix corresponds to the macroscopic effective tensor evaluated by the EBHM (line 48 in ConTop2D and line 45 in ConTop3D). Hence, the micro information needs to be considered in the optimization of the macrostructure.

Meanwhile, the first-order sensitivities of the structural compliance and micro volume fractions with respect to the micro

element densities are calculated in lines 57 to 66 (ConTop2D) and lines 54 to 66 (ConTop3D). We can find that the derivatives of the homogenized elastic tensor with respect to micro design variables are employed to compute the sensitivities of the macro element stiffness matrix with respect to micro element densities (ConTop2D: lines 59 to 62 and ConTop3D: lines 56 to 62). Then, the micro sensitivities of the objective function are obtained by substituting (10) into (9) and implemented in the codes (ConTop2D: lines 63 to 64 and ConTop3D: lines 63 to 64). As we can see, the macro displacement field needs to be considered in the computation of the micro derivatives of the structural compliance, as shown in line 63 (ConTop2D and ConTop3D). Hence, the information at the macroscale also affect the updating of the microstructure.

```

52 % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
53 ce = reshape(sum((U(edofMat).*Ke).*U(edofMat),2),Macro.nely,Macro.nelx);
54 c = sum(sum((Emin+Macro.xPhys.^penal*(1-Emin)).*ce));
55 Macro.dc = -penal*(1-Emin)*Macro.xPhys.^ (penal-1).*ce;
56 Macro.dv = ones(Macro.nely, Macro.nelx);
57 Micro.dc = zeros(Micro.nely, Micro.nelx);
58 for i = 1:Micro.nele
59     dDHe = [dDH{1,1}(i) dDH{1,2}(i) dDH{1,3}(i);
60             dDH{2,1}(i) dDH{2,2}(i) dDH{2,3}(i);
61             dDH{3,1}(i) dDH{3,2}(i) dDH{3,3}(i)];
62     [dKE] = elementMatVec2D(Macro.Elex, Macro.Eley, dDHe);
63     dce = reshape(sum((U(edofMat).*dKE).*U(edofMat),2),Macro.nely,Macro.nelx);
64     Micro.dc(i) = -sum(sum((Emin+Macro.xPhys.^penal*(1-Emin)).*dce));
65 end
66 Micro.dv = ones(Micro.nely, Micro.nelx);
49 % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
50 ce =
reshape(sum((U(edofMat).*Ke).*U(edofMat),2),[Macro.nely,Macro.nelx,Macro.nelz]);
51 c = sum(sum((Emin+Macro.xPhys.^penal*(1-Emin)).*ce)));
52 Macro.dc = -penal*(1-Emin)*Macro.xPhys.^ (penal-1).*ce;
53 Macro.dv = ones(Macro.nely, Macro.nelx, Macro.nelz);
54 Micro.dc = zeros(Micro.nely, Micro.nelx, Micro.nelz);
55 for i = 1:Micro.nele
56     dDHe = [dDH{1,1}(i) dDH{1,2}(i) dDH{1,3}(i) dDH{1,4}(i) dDH{1,5}(i)
dDH{1,6}(i);
57             dDH{2,1}(i) dDH{2,2}(i) dDH{2,3}(i) dDH{2,4}(i) dDH{2,5}(i) dDH{2,6}(i);
58             dDH{3,1}(i) dDH{3,2}(i) dDH{3,3}(i) dDH{3,4}(i) dDH{3,5}(i) dDH{3,6}(i);
59             dDH{4,1}(i) dDH{4,2}(i) dDH{4,3}(i) dDH{4,4}(i) dDH{4,5}(i) dDH{4,6}(i);
60             dDH{5,1}(i) dDH{5,2}(i) dDH{5,3}(i) dDH{5,4}(i) dDH{5,5}(i) dDH{5,6}(i);
61             dDH{6,1}(i) dDH{6,2}(i) dDH{6,3}(i) dDH{6,4}(i) dDH{6,5}(i) dDH{6,6}(i)];
62     [dKE] = elementMatVec3D(Macro.Elex, Macro.Eley, Macro.Elez, dDHe);
63     dce =
reshape(sum((U(edofMat).*dKE).*U(edofMat),2),[Macro.nely,Macro.nelx,Macro.nelz]);
64     Micro.dc(i) = -sum(sum((Emin+Macro.xPhys.^penal*(1-Emin)).*dce)));
65 end
66 Micro.dv = ones(Micro.nely, Micro.nelx, Micro.nelz);

```

5 Numerical examples

In this section, several numerical examples are tested to present the effectiveness of the Matlab codes for the concurrent topology optimization. In all examples, the sizes of material microstructure in all normal directions are defined to be 0.1. The Young's moduli for the solids and voids are 1 and $1e-9$, respectively. The Poisson's ratio is 0.3, and the penalization parameter p is defined as 3. A personal computer with the Matlab 9.1.0.441655 (R2016b) is used in the following examples.

5.1 Cantilever beam

In this example, the cantilever beam in Fig. 8 is studied, where a force ($F = -1$) is loaded at the middle point of the right side. The length and width of the beam is defined as follows: $L = 10$ and $H = 5$. Assuming that the macrostructure is discretized into 100×50 four-node plane stress finite elements, 50×50 plane stress finite elements are used to discretize the micro-

structure. The maximum of the volume fraction V_M is defined as 40%, and V_m for the microstructure is equal to 50%.

5.1.1 Concurrent topology optimization design

The initial designs of the macrostructure and microstructure are defined in Fig. 9. The macro initial design is uniformly distributed by an identical element density to avoid the local minimum designs (Bendsøe and Sigmund 2003). However, the initial design of the microstructure should be filled with some voids so as to avoid a uniformly distributed sensitivity field due to the initial imposed periodic boundary conditions (Sigmund 1994; Xia and Breitkopf 2015; Gao et al. 2018a). In this example, the microstructural design domain contains a hole to enable the inhomogeneous distribution of materials, which has been employed in many previous works (Huang et al. 2013; Xia and Breitkopf 2015; Gao et al. 2018a), and it is not a unique case for the current example. The concurrent topology optimization for the cantilever beam is performed by the following Matlab calling:

```

1 Macro_struct = [10, 5, 100, 50, 0.4];
2 Micro_struct = [0.1, 0.1, 50, 50, 0.5]; penal = 3; rmin = 2;
3 ConTop2D(Macro_struct, Micro_struct, penal, rmin)

```

Fig. 9 Initial designs at two scales. **a** The macrostructure. **b** The microstructure

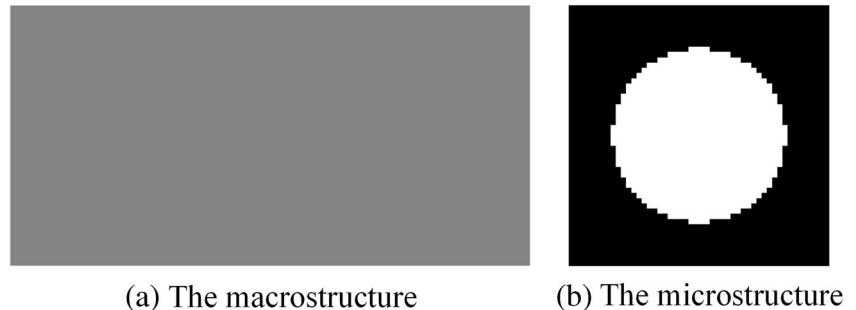


Table 3 The optimized results of the cantilever beam

The concurrent design of the cantilever beam	Homogenized elastic tensor	J
	$\begin{bmatrix} 0.368 & 0.073 & 0 \\ 0.073 & 0.107 & 0 \\ 0 & 0 & 0.069 \end{bmatrix}$	292.98

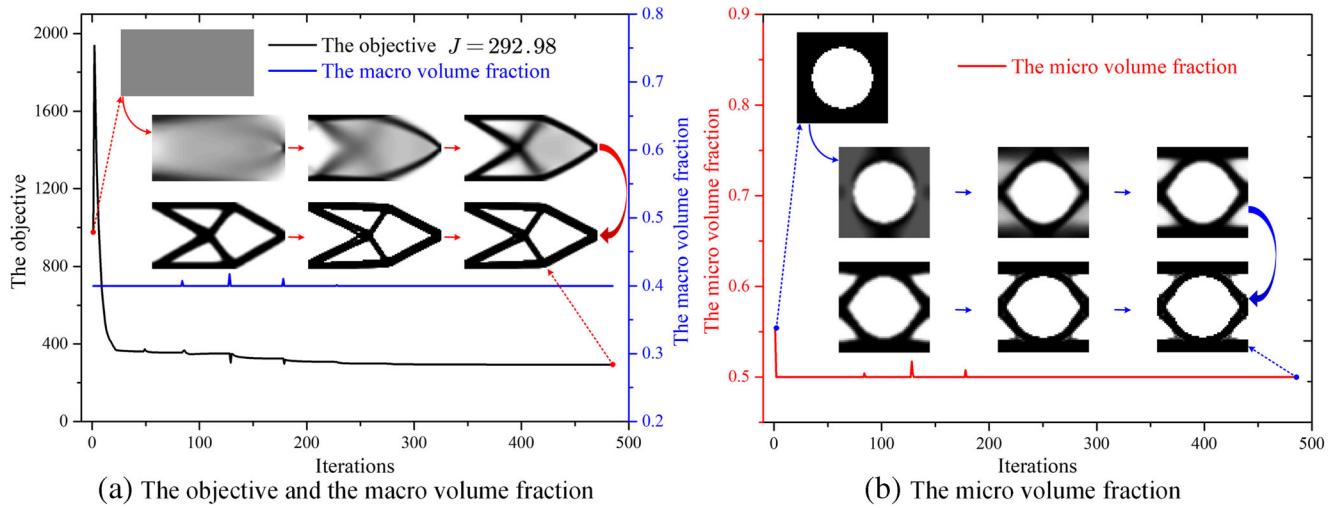
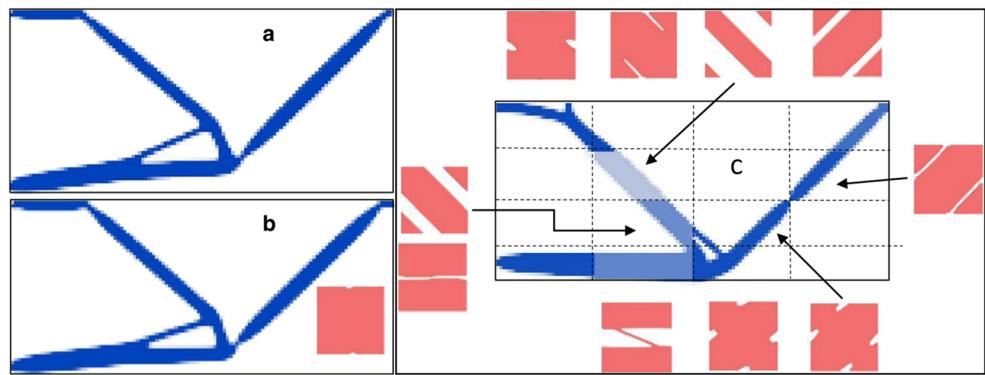


Fig. 10 Iterative histories. **a** The objective and the macro volume fraction. **b** The micro volume fraction

Table 4 The optimized results of the cantilever beam in six cases

Case 1	1 Macro_struct = [10, 5, 100, 50, 0.2];			$J_1 = 161.19$ $\begin{cases} V_M = 0.2 \\ V_m = 1 \end{cases}$
	2 Micro_struct = [0.1, 0.1, 50, 50, 1];			
Case 2	1 Macro_struct = [10, 5, 100, 50, 0.3];			$J_2 = 242.67$ $\begin{cases} V_M = 0.3 \\ V_m = 0.2/0.3 \end{cases}$
	2 Micro_struct = [0.1, 0.1, 50, 50, 0.2/0.3];			
Case 3	1 Macro_struct = [10, 5, 100, 50, 0.5];			$J_3 = 323.32$ $\begin{cases} V_M = 0.5 \\ V_m = 0.4 \end{cases}$
	2 Micro_struct = [0.1, 0.1, 50, 50, 0.4];			
Case 4	1 Macro_struct = [10, 5, 100, 50, 0.6];			$J_4 = 366.95$ $\begin{cases} V_M = 0.6 \\ V_m = 0.2/0.6 \end{cases}$
	2 Micro_struct = [0.1, 0.1, 50, 50, 0.2/0.6];			
Case 5	1 Macro_struct = [10, 5, 100, 50, 0.8];			$J_5 = 448.75$ $\begin{cases} V_M = 0.8 \\ V_m = 0.2/0.8 \end{cases}$
	2 Micro_struct = [0.1, 0.1, 50, 50, 0.2/0.8];			
Case 6	1 Macro_struct = [10, 5, 100, 50, 1];			$J_6 = 648.29$ $\begin{cases} V_M = 1 \\ V_m = 0.2 \end{cases}$
	2 Micro_struct = [0.1, 0.1, 50, 50, 0.2];			

Fig. 11 Concurrent topological design of the compliant mechanism (Sivapuram et al. 2016). **a** single scale. **b** uniform microstructure. **c** concurrent design with 12 microstructures



As listed in Table 3, the optimized results of the cantilever beam are provided, including the topology of the macrostructure, the topology of material microstructure and its homogenized elastic tensor, the concurrent design, and the optimized objective function. The optimized microtopology is very similar to the result of the microdesign in Huang et al. (2013), in which the macro topology keeps unchanged and its aim is to study the effect of the macroloads and boundary conditions on the micro optimization. As already pointed out in Kato et al. (2014), the structural responses, namely the stress distribution in the horizontal, vertical, and shear directions, determine the material layout in the microstructure. In this example, a slight difference is existing compared to the work in Kato et al. (2014). The optimized topology of the cantilever beam here has the symmetric distribution of the shear stress due to the load imposed at the middle of the left edge. Hence, a symmetric distribution of materials with nearly 45° and -45° is reasonable in order to against shear deformation of the macrostructure, and the material in the microstructure is also distributed in the horizontal direction to bear the horizontal stress in the macro design. We can confirm that the provided 2D Matlab code can implement the macro and micro designs to improve the structural performance.

Additionally, the macro design in the concurrent topology optimization is still analogous to the single-scale design for the macrostructure, shown in (Liu et al. 2008; Xia and Breitkopf 2014, 2017; Yan et al. 2014; Wang et al. 2016; Sivapuram et al. 2016; Chen et al. 2017, Xu and Cheng 2018; Gao et al. 2019a, b). The macro loads and boundary conditions might be the dominant factors to determine the topologies of both the macrostructure and microstructure in the concurrent topology optimization. It is reasonable for the macrostructure to have the analogous topologies, owing to the same loads and boundary conditions between the concurrent topology optimization and the single-scale optimization. A slight difference is also existed resulting from a fact that the spatial arrangement of the optimized material microstructure in the macrostructure affect the macro topology to some extent, particularly for the concurrent topology optimization considering multiple kinds of material microstructures (Xia and Breitkopf 2014, 2017; Wang et al. 2017a, 2018b; Xu and Cheng 2018; Gao et al. 2019a, b). Hence, the concurrent topology optimization aims to enlarge the possible design space for further improving the structural performance by adding the additional design freedoms from material microstructures, compared to the single-scale optimization.

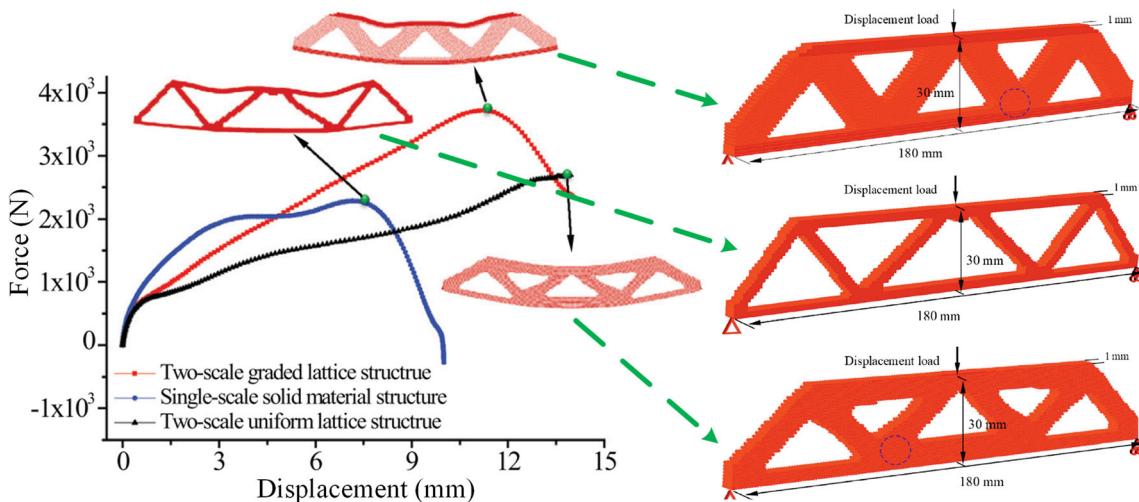


Fig. 12 Force-displacement curves for three designs in (Zhang et al. 2019)

Meanwhile, we present the iterations in detail for the objective function and volume fractions at two scales in Fig. 10. It can be easily seen that the iterations for the concurrent topology optimization are convergent in a stable process, owing to the perfect characteristics of the well-known SIMP approach. Meanwhile, some sharp jumps are occurred, due to the adoption of the Heaviside projection filter to ensure the approximate black-white design.

5.1.2 Influence of the volume fractions

In this subsection, we discuss the influence of the macro- and micro volume fractions on the structural performance based on the 2D Matlab code. Six different combinations of the macro- and micro volume fractions are defined, where macrovolume fractions are 0.2, 0.3, 0.5, 0.6, 0.8, and 1, respectively. The total volume fraction keeps unchanged, and micro volume fractions are equal to 1, 0.2/0.3, 0.4, 0.2/0.6, 0.2/0.8, and 0.2, respectively. Other parameters keep consistent with Sect. 5.1.1. It should be noted that the developed formulation in (4) can naturally degrade to the single-scale optimization if one of the macro- and micro volume fractions is set as 1. That is, when the micro volume fraction is equal to 1, the microstructure is fully filled with materials, namely the solid material microstructure. If the macro volume fraction is equal to 1, only the micro topology is optimized with the consideration of the effect of the macro loads and boundary conditions.

By calling the Matlab codes in Table 4, the concurrent designs of the cantilever beam in six cases are obtained and listed in Table 4, respectively. Similar to Table 3, the topologies of the macro- and microscales, the concurrent designs, and the optimized objective functions are included. As we can see, the optimized structural mean compliance gradually increases, with the increasing of the volume fraction of the macrostructure, namely $J_1 < J_2 < J_3 < J_4 < J_5 < J_6$. The design with the solid microstructure in case 1 has the best stiffness performance. Hence, the optimal design for the cantilever beam is the single-scale topology optimization of the macro-

structure in terms of the current example. The similar conclusions can also be obtained in (Vicente et al. 2016; Sivapuram et al. 2016), where the minimum compliance with a single load is considered.

As displayed in Fig. 11, two concurrent topology optimization designs are provided to present better mechanical advantage than the single-scale design (Sivapuram et al. 2016). In Yan et al. (2015), the two-scale design of multiscale composite structures with thermal insulation materials is featured with the better performance. Zhang et al. (2019) found the multiscale composite structures have the better capability to absorb the energies, which are displayed in Fig. 12. Overall, porous composite materials can offer the specific properties and have received enormous attentions and applications (Lee et al. 2012; Zheng et al. 2014; Gu et al. 2017; Yu et al. 2018). Hence, the concurrent topology optimization of the multiscale composite structures can be generalized to a broad of the topological design problems, not limited to the stiffness design.

Finally, it should be noted that the concurrent topology optimization formulation in (4) considers two volume constraints for the macrostructure and microstructure. The optimization with a total volume constraint can also be developed, similar to Sivapuram et al. 2016, where the optimizer can seek the optimal volume fractions during the optimization. Here, it is easier to briefly discuss the influence of the volume fractions at two scales, using the formulation with two volume constraints in (4). The formulation can also be viewed as a basis for concurrent topology optimization problems with multiple kinds of material microstructures.

5.1.3 Influence of the loads

In this subsection, the force in Fig. 8 is re-loaded at the bottom point of the right side to discuss the influence of the macro loads and boundary conditions on the micro optimization. The parameters are same as Sect. 5.1.1, including macro, micro, penal and rmin. The macroload condition needs to be modified in the line 13 of the main program (ConTop2D):

```
1 [load x, load y] = meshgrid(Macro.nelx, 0);
```

As shown in Table 5, the optimized topologies of the macrostructure and material microstructure, the concurrent design, and the homogenized elastic tensor of material microstructure are provided. As we can see, the optimized microstructure is featured with the anisotropy, due to the non-symmetry of the macro loads and boundary conditions.

The current design of the microstructure is similar to the result of the micro in Kato et al. (2014). The related discussions about the effect of the mechanical responses on the micro optimization to obtain the mechanically reasonable topology of material microstructure can also refer to Kato et al. (2014).

5.1.4 Influence of the initial design at the micro

In this subsection, the influence of the micro initial design on the optimization is addressed. As shown in Fig. 13, we define two different initial designs of material microstruc-

ture. A similar feature is that the micro design domain is homogeneously filled with holes. The Matlab codes in lines 32 to 38 of the main program (ConTop2D) for the micro initial design should be modified in two cases.

In case 1, the modified Matlab codes are given as follows:

```

for i = 1:Micro.nelx
    for j = 1:Micro.nely
        for k = [0 1]
            for l = [0 1]
                if sqrt((i-Micro.nelx*k-0.5)^2+(j-Micro.nely*l-0.5)^2) <
min(Micro.nelx, Micro.nely)/4
                    Micro.x(j,i) = 0;
                end
            end
        end
    end
end

```

The initial Matlab codes in lines 32 to 38 should be replaced for case 2 by the following:

```

for i = 1:Micro.nelx
    for j = 1:Micro.nely
        for k = [0 1]
            for l = [0 1]
                if sqrt((i-Micro.nelx*k-0.5)^2+(j-Micro.nely*l-0.5)^2) <
min(Micro.nelx, Micro.nely)/5
                    Micro.x(j,i) = 0;
                end
            end
        end
    end
    for k = 1/2
        for l = 1/2
            if sqrt((i-Micro.nelx*k-0.5)^2+(j-Micro.nely*l-0.5)^2) <
min(Micro.nelx, Micro.nely)/5
                Micro.x(j,i) = 0;
            end
        end
    end
end
end

```

The optimized results of two cases are provided in Table 6, including the concurrent designs of the cantilever beam, the homogenized elastic tensors of the optimized microstructures, and the optimized objective functions. It can be found that the optimized structural compliances in two cases are nearly identical, and also mostly equal to Sect. 5.1.1, although the topologies of material microstructure are different. It mainly

results from the non-uniqueness of the optimization of material microstructure (Sigmund 1994; Xia and Breitkopf 2015; Gao et al. 2018a, b). As given in the third column of Table 6, the homogenized elastic tensors in two cases are mostly close, and the optimized macro topologies in two cases are similar. Hence, we can confirm that the initial design of the micro has a negligible effect on the macro optimization.

Table 5 The optimized results of the cantilever beam

```

1 Macro_struct = [10, 5, 100, 50, 0.4];
2 Micro_struct = [0.1, 0.1, 50, 50, 0.5];
3 penal = 3; rmin = 2;
4 ConTop2D(Macro_struct, Micro_struct, penal, rmin)

```

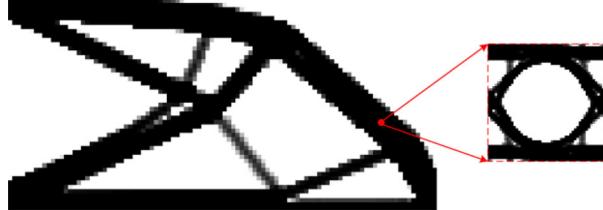
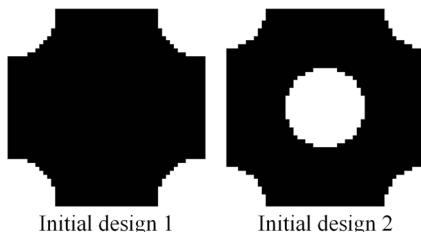
The concurrent design of the cantilever beam	Homogenized elastic tensor
	$\begin{bmatrix} 0.352 & 0.076 & -0.006 \\ 0.076 & 0.150 & -0.008 \\ -0.006 & -0.008 & 0.069 \end{bmatrix}$
 Initial design 1 Initial design 2	

Fig. 13 Two initial designs of material microstructure**Table 6** The optimized results in two cases

Case	Concurrent design of the cantilever beam	Homogenized elastic tensor	J
1		$\begin{bmatrix} 0.366 & 0.072 & 0 \\ 0.072 & 0.080 & 0 \\ 0 & 0 & 0.081 \end{bmatrix}$	293.83
2		$\begin{bmatrix} 0.374 & 0.070 & 0 \\ 0.070 & 0.095 & 0 \\ 0 & 0 & 0.082 \end{bmatrix}$	291.68

Fig. 14 The concurrent design of the MBB beam

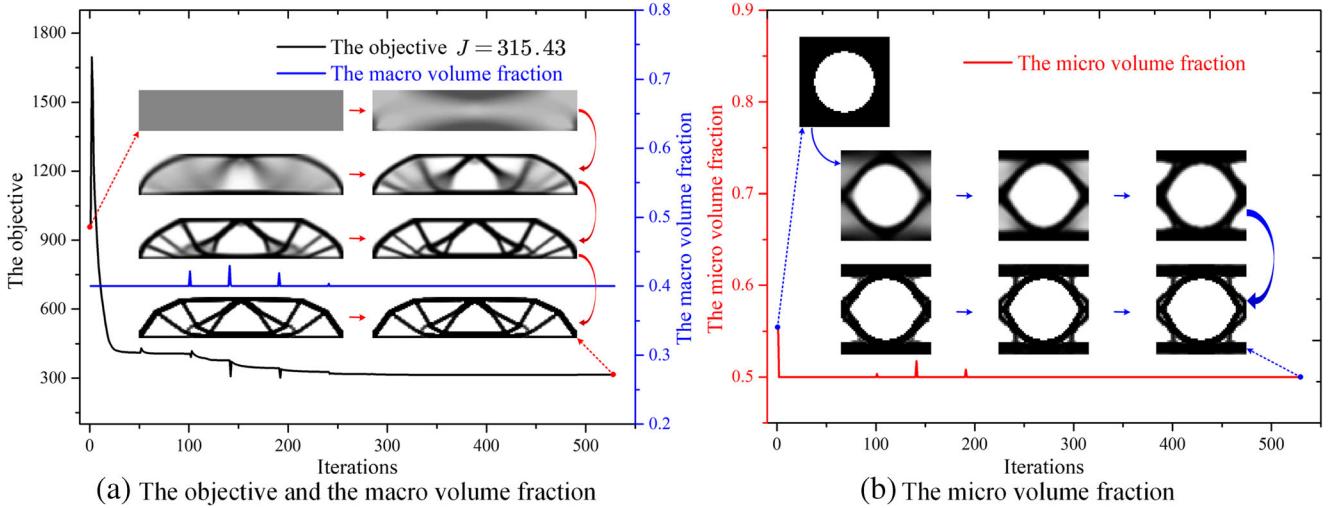


Fig. 15 Iterative histories. **a** The objective and the macro volume fraction. **b** The micro volume fraction

5.2 MBB beam

In this example, the well-known MBB beam is optimized to further show the effectiveness of the 2D code. A force is loaded at the middle point of the top edge, and structural sizes are defined as follows: $L = 15$ and $H = 3$. Four-node plane stress finite

elements (150×30) are utilized to discretize the macrostructure, and the microstructure is discretized into 50×50 plane stress finite elements. The maximums in the macro and micro volume constraints are defined as 40% and 50%, respectively. The macro loads and boundary conditions in ConTop2D main program should be replaced by the following:

```
1 [load_x, load_y] = meshgrid(Macro.nelx/2, Macro.nely);
2 [fixed_x, fixed_y] = meshgrid([0 Macro.nelx], 0);
3 fixeddofs = [2*fixednid(); 2*fixednid(1)-1];
```

and the corresponding main program is called by the following lines:

```
1 Macro_struct = [15, 3, 150, 30, 0.4];
2 Micro_struct = [0.1, 0.1, 50, 50, 0.5];
3 penal = 3; rmin = 2;
4 ConTop2D(Macro_struct, Micro_struct, penal, rmin)
```

The concurrent design of the MBB beam is shown in Fig. 14. It can be easily seen that the optimized design of the MBB beam by the current Matlab code is similar to the previous works (Yan et al. 2014; Wang et al. 2016; Chen et al. 2017), which demonstrates the effectiveness of the provided 2D Matlab code. Meanwhile, the iterative curves of the objective function, the macro and micro volume fractions are shown in Fig. 15, where the intermediate designs at two scales are also presented.

5.3 Michell-type structure

In this example, the 2D Matlab code is also applied to optimize the classic Michell-type structure to demonstrate its utility. The structural design domain is consistent with the first example shown in Fig. 8, while the macro loads and boundary

conditions are different. A force is loaded at the middle point of the bottom edge. The right-bottom corner is fixed, while the left-bottom corner is supported on a roller. The normal sizes are set as follows: $L = 10$ and $H = 4$. Four-node plane stress finite elements (100×40) are utilized in the numerical analysis, while the microscale finite mesh keeps unchanged.



Fig. 16 The concurrent design of the Michell-type structure

Meanwhile, in the main function ConTop2D, the macro loads and boundary conditions are re-defined as follows:

```

1 [load_x, load_y] = meshgrid(Macro.nelx/2, 0);
2 [fixed_x, fixed_y] = meshgrid([0 Macro.nelx], 0);
3 fixedddofs = [2*fixednid(:); 2*fixednid(1)-1];

```

and the corresponding main function is called by the following lines:

```

1 Macro_struct = [10, 4, 100, 40, 0.4];
2 Micro_struct = [0.1, 0.1, 50, 50, 0.5];
3 penal = 3; rmin = 2;
4 ConTop2D(Macro_struct, Micro_struct, penal, rmin)

```

As plotted in Fig. 16, the final concurrent topology optimization design solution is provided. Similarly, the stiffness performance of the Michell-type structure is also optimized considering from two basic elements, namely the macrostructure and material microstructure. The optimized microstructure is uniformly distributed in the final design at the macroscale. Meanwhile, the convergent trajectories of the objective function, the macro volume fraction, and micro volume fraction are displayed in Fig. 17, where some intermediate designs are also provided.

5.4 3D structures

In this example, we intend to demonstrate the effectiveness of the 3D Matlab code. The macro loads and boundary conditions are defined in the design domain for the 3D-supported structure, shown in Fig. 18. Four corners at the bottom surface are fixed, and a unit downward force is loaded at the middle of the top surface. The normal dimensions of the macrostructure are $L=2$, $W=1.6$, and $H=2$, which is discretized by $20 \times 16 \times 20$ finite elements. The microstructure is discretized into $20 \times 20 \times 20$ finite elements. The volume constraints at two scales are defined as 20% and 30%, respectively. The final concurrent topology optimization design of the 3D-supported structure, as displayed in Fig. 19, can be obtained by means of the following function call:

```

1 Macro_struct = [2, 1.6, 2, 20, 16, 20, 0.2];
2 Micro_struct = [0.1, 0.1, 0.1, 20, 20, 20, 0.3];
3 penal = 3; rmin = 2.0;
4 ConTop3D(Macro_struct, Micro_struct, penal, rmin)

```

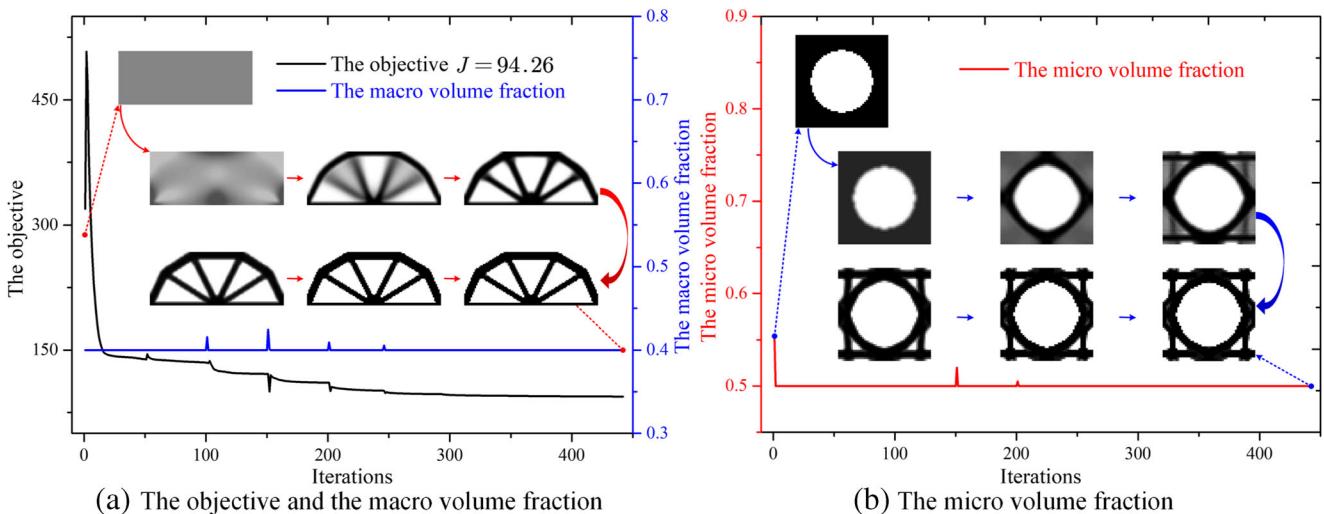


Fig. 17 Iterative histories. **a** The objective and macro volume fraction. **b** The micro volume fraction

It can be seen that the macro topology and micro topology are concurrently optimized to improve the structural stiffness performance of the 3D-supported structures. Hence, the effectiveness of the current Matlab code for 3D concurrent topology optimization problems can be demonstrated.

In terms of the 3D cantilever beam, the normal dimensions of the macrostructure are defined as 6, 2, and 0.4

```

1 [load_x,load_y,load_z] = meshgrid(Macro.nelx, Macro.nely/2, 0:Macro.nelz);
2 loadnid =
3 load_z*(Macro.nelx+1)*(Macro.nely+1)+load_x*(Macro.nely+1)+(Macro.nely+1-loa
d_y);
4 F = sparse(3*loadnid(:) - 1,1,-1,Macro.ndof,1);
5 U = zeros(Macro.ndof,1);
6 [fixed_x,fixed_y,fixed_z] = meshgrid(0,0:Macro.nely,0:Macro.nelz);
7 fixednid =
8 fixed_z*(Macro.nelx+1)*(Macro.nely+1)+fixed_x*(Macro.nely+1)+(Macro.nely+1-f
ixed_y);
9 fixeddofs = [3*fixednid(:); 3*fixednid(:)-1; 3*fixednid(:)-2];
10 freedofs = setdiff(1:Macro.ndof,fixeddofs);

```

The concurrent topology optimization design for the 3D cantilever beam can be obtained by calling the following lines. The final concurrent design solution is clearly demonstrated in Fig. 20. It can be easily seen that the final design with the optimized structural performance contains two design pillars, which further show the effectiveness of the 3D Matlab code.

```

1 Macro_struct = [2, 1.6, 2, 20, 16, 20, 0.2];
2 Micro_struct = [0.1, 0.1, 0.1, 20, 20, 20, 0.3];
3 penal = 3; rmin = 2.0;
4 ConTop3D(Macro_struct, Micro_struct, penal, rmin)

```

6 Conclusions

This paper provides 2D and 3D Matlab codes to implement the concurrent topology optimization for multiscale design of cellular composite structures. The modified SIMP approach combined with the EBHM is applied to implement the design. The Matlab implementations for the 3D EBHM are shown in detail, with the key component for how to develop the 3D periodic boundary formulation. The basic code architecture of the concurrent topology optimization is also developed, with the two-scale sensitivity analysis. Finally, several 2D and 3D numerical examples are fully discussed to show the effectiveness of the Matlab codes attached in the [Appendix](#). Although the benchmark stiffness design is used as numerical examples due to its easiness and wide acceptance in the field, the multiscale composite structures might be more beneficial for a broad range of advanced designs and applications, beyond the conventional single-scale solid design.

respectively, and $60 \times 20 \times 4$ finite elements are applied to discretize the macro design domain. The micro mesh keeps consistent with the example for the 3D-supported structure. Firstly, the macro boundary and load conditions (lines 13 to 20 in ConTop3D) are re-defined for the 3D cantilever beam by the following Matlab lines:

However, it is noted that the two 3D examples are only demonstrative, with a coarse finite element mesh to reduce the computational cost. The multigrid preconditioned conjugate gradient solver (Amir et al. 2014) can be used to improve the numerical efficiency in the 3D multiscale composites.

7 Replication of results

The authors reserve all rights for the programs. The programs may be distributed and used for academic and educational purposes. The authors do not guarantee that the code is free from errors, and they shall not be liable in any event caused by the use of the program.

Funding information This work was partially supported by the Australian Research Council (ARC)—Discovery Projects (160102491), the National Basic Scientific Research Program of China (JCKY2016110C012), and the National Natural Science Foundation of China (51705165).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

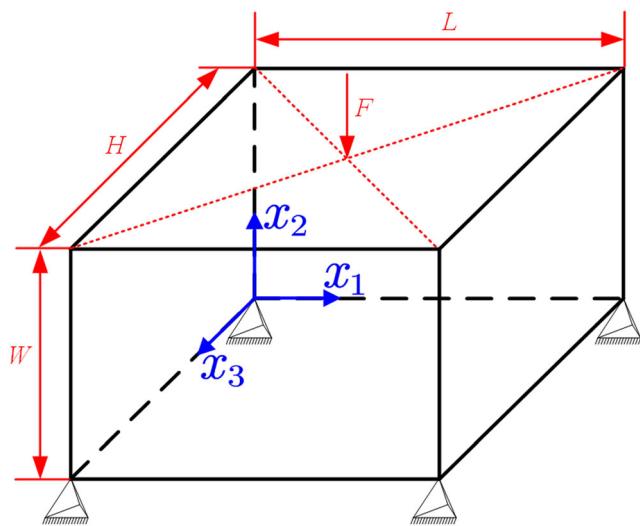


Fig. 18 A 3D-supported structure

Fig. 19 The concurrent design of the 3D-supported structure

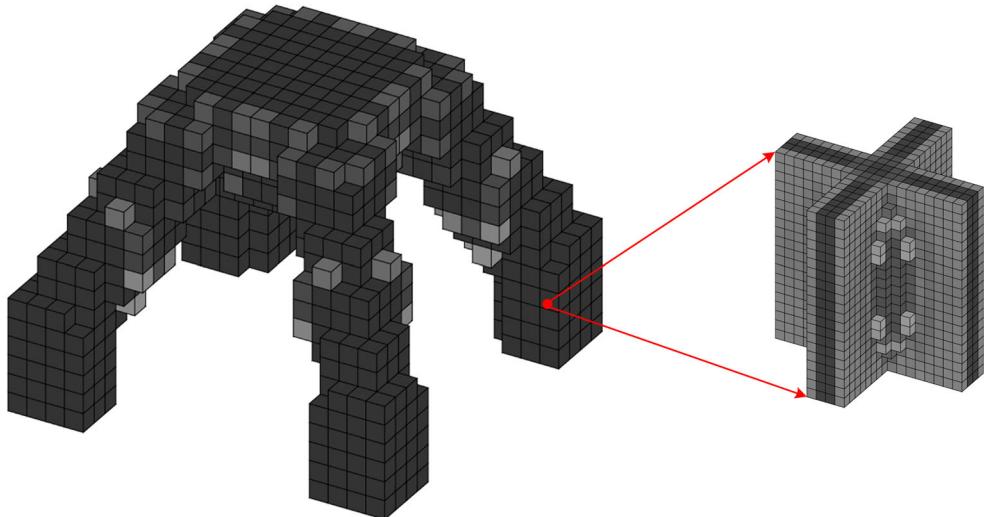
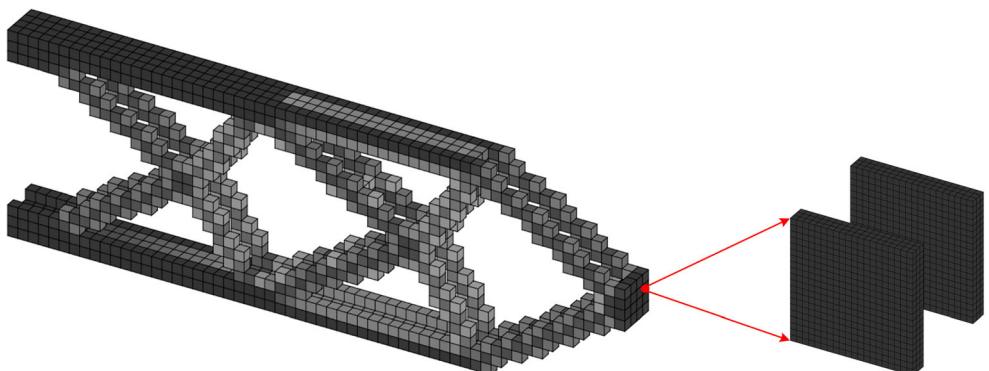


Fig. 20 The concurrent design of the 3D cantilever beam



Appendix. MATLAB codes for concurrent topology optimization in 2D and 3D

ConTop2D

```

1 function ConTop2D(Macro_struct, Micro_struct, penal, rmin)
2 % USER-DEFINED DESIGN AND STRUCTURAL PARAMETERS
3 maxloop = 200; E0 = 1; Emin = 1e-9; nu = 0.3;
4 Macro.length = Macro_struct(1); Macro.width = Macro_struct(2);
5 Micro.length = Micro_struct(1); Micro.width = Micro_struct(2);
6 Macro.nelx = Macro_struct(3); Macro.nely = Macro_struct(4);
7 Micro.nelx = Micro_struct(3); Micro.nely = Micro_struct(4);
8 Macro.Vol = Macro_struct(5); Micro.Vol = Micro_struct(5);
9 Macro.Elex = Macro.length/Macro.nelx; Macro.Eley = Macro.width/Macro.nely;
10 Macro.nele = Macro.nelx*Macro.nely; Micro.nele = Micro.nelx*Micro.nely;
11 Macro.ndof = 2*(Macro.nelx+1)*(Macro.nely+1);
12 % PREPARE FINITE ELEMENT ANALYSIS
13 [load_x, load_y] = meshgrid(Macro.nelx, Macro.nely/2);
14 loadnid = load_x*(Macro.nely+1)+(Macro.nely+1-load_y);
15 F = sparse(2*loadnid(:, 1, -1, 2*(Macro.nelx+1)*(Macro.nely+1), 1);
16 U = zeros(Macro.ndof,1);
17 [fixed_x, fixed_y] = meshgrid(0, 0:Macro.nely);
18 fixednid = fixed_x*(Macro.nely+1)+(Macro.nely+1-fixed_y);
19 fixeddofs = [2*fixednid(:, 2*fixednid(:, -1];
20 freedofs = setdiff(1:Macro.ndof,fixeddofs);
21 nodenrs =
    reshape(1:(Macro.nely+1)*(Macro.nelx+1),1+Macro.nely,1+Macro.nelx);
22 edofVec = reshape(2*nodenrs(1:end-1,1:end-1)+1,Macro.nele,1);
23 edofMat = repmat(edofVec,1,8)+repmat([0 1 2*Macro.nely+[2 3 0 1] -2
    -1],Macro.nele,1);
24 iK = reshape(kron(edofMat,ones(8,1))',64*Macro.nele,1);
25 jK = reshape(kron(edofMat,ones(1,8))',64*Macro.nele,1);
26 % PREPARE FILTER
27 [Macro.H,Macro.Hs] = filtering2d(Macro.nelx, Macro.nely, Macro.nele, rmin);
28 [Micro.H,Micro.Hs] = filtering2d(Micro.nelx, Micro.nely, Micro.nele, rmin);
29 % INITIALIZATIONS AT TWO SCALES
30 Macro.x = repmat(Macro.Vol,Macro.nely,Macro.nelx);
31 Micro.x = ones(Micro.nely,Micro.nelx);
32 for i = 1:Micro.nelx
33     for j = 1:Micro.nely
34         if sqrt((i-Micro.nelx/2-0.5)^2+(j-Micro.nely/2-0.5)^2) <
            min(Micro.nelx,Micro.nely)/3
35             Micro.x(j,i) = 0;
36         end
37     end
38 end
39 beta = 1;
40 Macro.xTilde = Macro.x; Micro.xTilde = Micro.x;
41 Macro.xPhys = 1-exp(-beta*Macro.xTilde)+Macro.xTilde*exp(-beta);
42 Micro.xPhys = 1-exp(-beta*Micro.xTilde)+Micro.xTilde*exp(-beta);
43 loopbeta = 0; loop = 0; Macro.change = 1; Micro.change = 1;
44 while loop < maxloop || Macro.change > 0.01 || Micro.change > 0.01
45     loop = loop+1; loopbeta = loopbeta+1;
46     % FE-ANALYSIS AT TWO SCALES
47     [DH, dDH] = EBHM2D(Micro.xPhys, Micro.length, Micro.width, E0, Emin, nu,
        penal);
48     Ke = elementMatVec2D(Macro.Elex/2, Macro.Eley/2, DH);
49     SK =
        reshape(Ke(:)*(Emin+Macro.xPhys(:)').^penal*(1-Emin)),64*Macro.nele,1);
50     K = sparse(iK,jK,SK); K = (K+K')/2;
51     U(freedofs,:) = K(freedofs,freedofs)\F(freedofs,:);

```

```

52      % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
53      ce = reshape(sum((U(edofMat).*Ke).*U(edofMat),2),Macro.nely,Macro.nelx);
54      c = sum(sum((Emin+Macro.xPhys.^penal*(1-Emin)).*ce));
55      Macro.dc = -penal*(1-Emin)*Macro.xPhys.^ (penal-1).*ce;
56      Macro.dv = ones(Macro.nely, Macro.nelx);
57      Micro.dc = zeros(Micro.nely, Micro.nelx);
58      for i = 1:Micro.nele
59          dDHe = [dDH{1,1}(i) dDH{1,2}(i) dDH{1,3}(i);
60                  dDH{2,1}(i) dDH{2,2}(i) dDH{2,3}(i);
61                  dDH{3,1}(i) dDH{3,2}(i) dDH{3,3}(i)];
62          [dKE] = elementMatVec2D(Macro.Elex, Macro.Eley, dDHe);
63          dce =
64          reshape(sum((U(edofMat).*dKE).*U(edofMat),2),Macro.nely,Macro.nelx);
65          Micro.dc(i) = -sum(sum((Emin+Macro.xPhys.^penal*(1-Emin)).*dce));
66      end
67      % FILTERING AND MODIFICATION OF SENSITIVITIES
68      Macro.dx = beta*exp(-beta*Macro.xTilde)+exp(-beta); Micro.dx =
69      beta*exp(-beta*Micro.xTilde)+exp(-beta);
70      Macro.dc(:) = Macro.H*(Macro.dc(:).*Macro.dx(:)./Macro.Hs); Macro.dv(:)
71      = Macro.H*(Macro.dv(:).*Macro.dx(:)./Macro.Hs);
72      Micro.dc(:) = Micro.H*(Micro.dc(:).*Micro.dx(:)./Micro.Hs); Micro.dv(:)
73      = Micro.H*(Micro.dv(:).*Micro.dx(:)./Micro.Hs);
74      % OPTIMALITY CRITERIA UPDATE FOR MACRO AND MICRO ELEMENT DENSITIES
75      [Macro.x, Macro.xPhys, Macro.change] = OC(Macro.x, Macro.dc, Macro.dv,
76      Macro.H, Macro.Hs, Macro.Vol, Macro.nele, 0.2, beta);
77      [Micro.x, Micro.xPhys, Micro.change] = OC(Micro.x, Micro.dc, Micro.dv,
78      Micro.H, Micro.Hs, Micro.Vol, Micro.nele, 0.2, beta);
79      Macro.xPhys = reshape(Macro.xPhys, Macro.nely, Macro.nelx); Micro.xPhys
80      = reshape(Micro.xPhys, Micro.nely, Micro.nelx);
81      % PRINT RESULTS
82      fprintf(' It.:%5i Obj.:%11.4f Macro_Vol.:%7.3f Micro_Vol.:%7.3f
83      Macro_ch.:%7.3f Micro_ch.:%7.3f\n',...
84      loop,c,mean(Macro.xPhys(:)),mean(Micro.xPhys(:)), Macro.change,
85      Micro.change);
86      colormap(gray); imagesc(1-Macro.xPhys); caxis([0 1]); axis equal; axis
87      off; drawnow;
88      colormap(gray); imagesc(1-Micro.xPhys); caxis([0 1]); axis equal; axis
89      off; drawnow;
90      % UPDATE HEAVISIDE REGULARIZATION PARAMETER
91      if beta < 512 && (loopbeta >= 50 || Macro.change <= 0.01 || Micro.change
92      <= 0.01 )
93          beta = 2*beta; loopbeta = 0; Macro.change = 1; Micro.change = 1;
94      fprintf('Parameter beta increased to %g.\n',beta);
95      end
96  end
97 end

```

ConTop3D

```

1 function ConTop3D(Macro_struct, Micro_struct, penal, rmin)
2 % USER-DEFINED LOOP PARAMETERS
3 maxloop = 200; displayflag = 1; E0 = 1; Emin = 1e-9; nu = 0.3;
4 Macro.length = Macro_struct(1); Macro.width = Macro_struct(2); Macro.Height =
5 Macro_struct(3);
6 Micro.length = Micro_struct(1); Micro.width = Micro_struct(2); Micro.Height =
7 Micro_struct(3);
8 Macro.nelx = Macro_struct(4); Macro.nely = Macro_struct(5); Macro.nelz =
9 Macro_struct(6);
10 Micro.nelx = Micro_struct(4); Micro.nely = Micro_struct(5); Micro.nelz =
11 Micro_struct(6);

```

```

8 Macro.Vol = Macro_struct(7); Micro.Vol = Micro_struct(7);
9 Macro.Elex = Macro.length/Macro.nelx; Macro.Eley = Macro.width/Macro.nely;
Macro.Elez = Macro.Height/Macro.nelz;
10 Macro.nele = Macro.nelx*Macro.nely*Macro.nelz; Micro.nele =
Micro.nelx*Micro.nely*Micro.nelz;
11 Macro.ndof = 3*(Macro.nelx+1)*(Macro.nely+1)*(Macro.nelz+1);
12 % PREPARE FINITE ELEMENT ANALYSIS
13 [load_x,load_y,load_z] = meshgrid(Macro.nelx/2, Macro.nely, Macro.nelz/2);
14 loadnid =
load_z*(Macro.nelx+1)*(Macro.nely+1)+load_x*(Macro.nely+1)+(Macro.nely+1-loa
d_y);
15 F = sparse(3*loadnid(:) - 1,1,-1,Macro.ndof,1);
16 U = zeros(Macro.ndof,1);
17 [fixed_x,fixed_y,fixed_z] = meshgrid([0 Macro.nelx],0,[0 Macro.nelz]);
18 fixednid =
fixed_z*(Macro.nelx+1)*(Macro.nely+1)+fixed_x*(Macro.nely+1)+(Macro.nely+1-f
ixed_y);
19 fixeddofs = [3*fixednid(:); 3*fixednid(:)-1; 3*fixednid(:)-2];
20 freedofs = setdiff(1:Macro.ndof,fixeddofs);
21 nodegrd = reshape(1:(Macro.nely+1)*(Macro.nelx+1),Macro.nely+1,Macro.nelx+1);
22 nodeids = reshape(nodegrd(1:end-1,1:end-1),Macro.nely*Macro.nelx,1);
23 nodeidz =
0:(Macro.nely+1)*(Macro.nelx+1):(Macro.nelz-1)*(Macro.nely+1)*(Macro.nelx+1)
;
24 nodeids = repmat(nodeids,size(nodeidz))+repmat(nodeidz,size(nodeids));
25 edofMat = repmat(3*nodeids(:)+1,1,24)+ repmat([0 1 2 3*Macro.nely + [3 4 5 0 1
2] -3 -2 -1 ...
26 3*(Macro.nely+1)*(Macro.nelx+1)+[0 1 2 3*Macro.nely + [3 4 5 0 1 2] -3 -2
-1],Macro.nele,1);
27 iK = reshape(kron(edofMat,ones(24,1))',24*24*Macro.nele,1);
28 jK = reshape(kron(edofMat,ones(1,24))',24*24*Macro.nele,1);
29 % PREPARE FILTER
30 [Macro.H,Macro.Hs] = filtering3d(Macro.nelx, Macro.nely, Macro.nelz,
Macro.nele, rmin);
31 [Micro.H,Micro.Hs] = filtering3d(Micro.nelx, Micro.nely, Micro.nelz,
Micro.nele, rmin);
32 % INITIALIZE ITERATION
33 Macro.x = repmat(Macro.Vol,[Macro.nely,Macro.nelx,Macro.nelz]);
34 Micro.x = ones(Micro.nely,Micro.nelx,Micro.nelz);
35 Micro.x(Micro.nely/2:Micro.nely/2+1,Micro.nelx/2:Micro.nelx/2+1,Micro.nelz/2
:Micro.nelz/2+1) = 0;
36 beta = 1;
37 Macro.xTilde = Macro.x; Micro.xTilde = Micro.x;
38 Macro.xPhys = 1-exp(-beta*Macro.xTilde)+Macro.xTilde*exp(-beta);
39 Micro.xPhys = 1-exp(-beta*Micro.xTilde)+Micro.xTilde*exp(-beta);
40 loopbeta = 0; loop = 0; Macro.change = 1; Micro.change = 1;
41 while loop < maxloop || Macro.change > 0.01 || Micro.change > 0.01
42     loop = loop+1; loopbeta = loopbeta+1;
43     % FE-ANALYSIS AT TWO SCALES
44     [DH, dDH] = EBHM3D(Micro.xPhys, Micro.length, Micro.width, Micro.Height,
E0, Emin, nu, penal);
45     Ke = elementMatVec3D(Macro.Elex/2, Macro.Eley/2, Macro.Elez/2, DH);
46     SK =
reshape(Ke(:)*(Emin+Macro.xPhys(:)').^penal*(1-Emin)),24*24*Macro.nele,1);
47     K = sparse(iK,jK,SK); K = (K+K')/2;
48     U(freedofs,:) = K(freedofs,freedofs)\F(freedofs,:);
49     % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
50     ce =
reshape(sum((U(edofMat)*Ke).*U(edofMat),2),[Macro.nely,Macro.nelx,Macro.nelz
]);

```

```

51      c = sum(sum((Emin+Macro.xPhys.^penal*(1-Emin)).*ce)));
52      Macro.dc = -penal*(1-Emin)*Macro.xPhys.^ (penal-1).*ce;
53      Macro.dv = ones(Macro.nely, Macro.nelx, Macro.nelz);
54      Micro.dc = zeros(Micro.nely, Micro.nelx, Micro.nelz);
55      for i = 1:Micro.nele
56          dDHe = [dDH{1,1}(i) dDH{1,2}(i) dDH{1,3}(i) dDH{1,4}(i) dDH{1,5}(i)
57          dDH{1,6}(i); dDH{2,1}(i) dDH{2,2}(i) dDH{2,3}(i) dDH{2,4}(i) dDH{2,5}(i)
58          dDH{2,6}(i); dDH{3,1}(i) dDH{3,2}(i) dDH{3,3}(i) dDH{3,4}(i) dDH{3,5}(i)
59          dDH{3,6}(i); dDH{4,1}(i) dDH{4,2}(i) dDH{4,3}(i) dDH{4,4}(i) dDH{4,5}(i)
60          dDH{4,6}(i); dDH{5,1}(i) dDH{5,2}(i) dDH{5,3}(i) dDH{5,4}(i) dDH{5,5}(i)
61          dDH{5,6}(i); dDH{6,1}(i) dDH{6,2}(i) dDH{6,3}(i) dDH{6,4}(i) dDH{6,5}(i)
62          dDH{6,6}(i)];
63          [dKE] = elementMatVec3D(Macro.Elex, Macro.Eley, Macro.Elez, dDHe);
64          dce =
65          reshape(sum((U(edofMat).*dKE).*U(edofMat),2),[Macro.nely,Macro.nelx,Macro.nel
66          z]);
67          Micro.dc(i) = -sum(sum((Emin+Macro.xPhys.^penal*(1-Emin)).*dce)));
68      end
69      Micro.dv = ones(Micro.nely, Micro.nelx, Micro.nelz);
70      % FILTERING AND MODIFICATION OF SENSITIVITIES
71      Macro.dx = beta*exp(-beta*Macro.xTilde)+exp(-beta); Micro.dx =
72      beta*exp(-beta*Micro.xTilde)+exp(-beta);
73      Macro.dc(:) = Macro.H*(Macro.dc(:).*Macro.dx(:)./Macro.Hs); Macro.dv(:) =
74      Macro.H*(Macro.dv(:).*Macro.dx(:)./Macro.Hs);
75      Micro.dc(:) = Micro.H*(Micro.dc(:).*Micro.dx(:)./Micro.Hs); Micro.dv(:) =
76      Micro.H*(Micro.dv(:).*Micro.dx(:)./Micro.Hs);
77      % OPTIMALITY CRITERIA UPDATE FOR MACRO ELEMENT DENSITIES
78      [Macro.x, Macro.xPhys, Macro.change] = OC(Macro.x, Macro.dc, Macro.dv,
79      Macro.H, Macro.Hs, Macro.Vol, Macro.nele, 0.02, beta);
80      [Micro.x, Micro.xPhys, Micro.change] = OC(Micro.x, Micro.dc, Micro.dv,
81      Micro.H, Micro.Hs, Micro.Vol, Micro.nele, 0.02, beta);
82      Macro.xPhys = reshape(Macro.xPhys, Macro.nely, Macro.nelx, Macro.nelz);
83      Micro.xPhys = reshape(Micro.xPhys, Micro.nely, Micro.nelx, Micro.nelz);
84      % PRINT RESULTS
85      fprintf(' It.:%5i Obj.:%11.4f Macro_Vol.:%7.3f Micro_Vol.:%7.3f
86      Macro_ch.:%7.3f Micro_ch.:%7.3f\n',...
87          loop,c,mean(Macro.xPhys(:)),mean(Micro.xPhys(:)), Macro.change,
88          Micro.change);
89      if displayflag, clf; display_3D(Macro.xPhys); end;
90      if displayflag, clf; display_3D(Micro.xPhys); end
91      % UPDATE HEAVISIDE REGULARIZATION PARAMETER
92      if beta < 512 && (loopbeta >= 50 || Macro.change <= 0.01 || Micro.change <=
93      0.01 )
94          beta = 2*beta; loopbeta = 0; Macro.change = 1; Micro.change = 1;
95          fprintf('Parameter beta increased to %g.\n',beta);
96      end
97  end
98 end

```

References

- Alexandersen J, Lazarov BS (2015) Topology optimisation of manufacturable microstructural details without length scale separation using a spectral coarse basis preconditioner. *Comput Methods Appl Mech Eng* 290:156–182
- Allaire G, Jouve F, Toader AM (2004) Structural optimization using sensitivity analysis and a level-set method. *J Comput Phys* 194: 363–393
- Amir O, Aage N, Lazarov BS (2014) On multigrid-CG for efficient topology optimization. *Struct Multidiscip Optim* 49:815–829
- Andreassen E, Clausen A, Schevenels M et al (2011) Efficient topology optimization in MATLAB using 88 lines of code. *Struct Multidiscip Optim* 43:1–16
- Bendsøe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. *Comput Methods Appl Mech Eng* 71:197–224
- Bendsøe MP, Sigmund O (2003) Topology optimization: theory, methods and applications. Springer-Verlag, Berlin
- Bendsøe MP, Sigmund O (1999) Material interpolation schemes in topology optimization. *Arch Appl Mech* 69:635–654
- Challis VJ (2010) A discrete level-set topology optimization code written in Matlab. *Struct Multidiscip Optim* 41:453–464
- Chen W, Tong L, Liu S (2017) Concurrent topology design of structure and material using a two-scale topology optimization. *Comput Struct* 178:119–128
- Chu S, Gao L, Xiao M, Li H (2019) Design of sandwich panels with truss cores using explicit topology optimization. *Compos Struct* 210:892–905
- Cook RD, Malkus DS, Plesha ME, Witt RJ (1974) Concepts and applications of finite element analysis. Wiley, New York
- Da DC, Cui XY, Long K, Li GY (2017) Concurrent topological design of composite structures and the underlying multi-phase materials. *Comput Struct* 179:1–14
- Du Z, Zhou X-Y, Picelli R, Kim HA (2018) Connecting microstructures for multiscale topology optimization with connectivity index constraints. *J Mech Des* 140:111417
- Gao J, Li H, Gao L, Xiao M (2018a) Topological shape optimization of 3D micro-structured materials using energy-based homogenization method. *Adv Eng Softw* 116:89–102
- Gao J, Li H, Luo Z et al (2018b) Topology optimization of micro-structured materials featured with the specific mechanical properties. *Int J Comput Methods* 15:1850144
- Gao J, Luo Z, Li H et al (2019a) Dynamic multiscale topology optimization for multi-regional micro-structured cellular composites. *Compos Struct* 211:401–417
- Gao J, Luo Z, Li H, Gao L (2019b) Topology optimization for multiscale design of porous composites with multi-domain microstructures. *Comput Methods Appl Mech Eng* 344:451–476
- Gao J, Xue H, Gao L, Luo Z (2019c) Topology optimization for auxetic metamaterials based on isogeometric analysis. *Comput Methods Appl Mech Eng* 352:211–36. <https://doi.org/10.1016/J.CMA.2019.04.021>
- Groen JP, Sigmund O (2018) Homogenization-based topology optimization for high-resolution manufacturable microstructures. *Int J Numer Methods Eng* 113:1148–1163
- Gu GX, Takaffoli M, Buehler MJ (2017) Hierarchically enhanced impact resistance of bioinspired composites. *Adv Mater* 29:1700060
- Guedes JMJ, Kikuchi N (1990) Preprocessing and postprocessing for materials based on the homogenization method with adaptive finite element methods. *Comput Methods Appl Mech Eng* 83:143–198
- Guo X, Zhang W, Zhong W (2014) Doing topology optimization explicitly and geometrically—a new moving morphable components based framework. *J Appl Mech* 81:081009
- Guo X, Zhao X, Zhang W et al (2015) Multi-scale robust design and optimization considering load uncertainties. *Comput Methods Appl Mech Eng* 283:994–1009
- Hashin Z (1983) Analysis of composite materials—a survey. *J Appl Mech* 50:481–505
- Huang X, Xie YM (2010) A further review of ESO type methods for topology optimization. *Struct Multidiscip Optim* 41:671–683
- Huang X, Zhou SW, Xie YM, Li Q (2013) Topology optimization of microstructures of cellular materials and composites for macrostructures. *Comput Mater Sci* 67:397–407
- Kato J, Yachi D, Kyoya T, Terada K (2018) Micro-macro concurrent topology optimization for nonlinear solids with a decoupling multiscale analysis. *Int J Numer Methods Eng* 113:1189–1213
- Kato J, Yachi D, Terada K, Kyoya T (2014) Topology optimization of micro-structure for composites applying a decoupling multi-scale analysis. *Struct Multidiscip Optim* 49:595–608
- Lee JH, Singer JP, Thomas EL (2012) Micro-/nanostructured mechanical metamaterials. *Adv Mater* 24:4782–4810
- Li H, Luo Z, Gao L, Walker P (2018a) Topology optimization for functionally graded cellular composites with metamaterials by level sets. *Comput Methods Appl Mech Eng* 328:340–364
- Li H, Luo Z, Gao L, Qin Q (2018b) Topology optimization for concurrent design of structures with multi-patch microstructures by level sets. *Comput Methods Appl Mech Eng* 331:536–561
- Li H, Luo Z, Zhang N et al (2016) Integrated design of cellular composites using a level-set topology optimization method. *Comput Methods Appl Mech Eng* 309:453–475
- Liu K, Tovar A (2014) An efficient 3D topology optimization code written in Matlab. *Struct Multidiscip Optim* 50:1175–1196
- Liu L, Yan J, Cheng G (2008) Optimum structure with homogeneous optimum truss-like material. *Comput Struct* 86:1417–1425
- Long K, Yuan PF, Xu S, Xie YM (2018) Concurrent topological design of composite structures and materials containing multiple phases of distinct Poisson's ratios. *Eng Optim* 50:599–614
- Luo Z, Wang MY, Tong L, Wang S (2007) Shape and topology optimization of compliant mechanisms using a parameterization level set method. *J Comput Phys* 227:680–705
- Luo Z, Wang MY, Wang SY, Wei P (2008) A level set-based parameterization method for structural shape and topology optimization Internet. *J Numer Methods Eng* 76:1–26
- Niu B, Yan J, Cheng G (2009) Optimum structure with homogeneous optimum cellular material for maximum fundamental frequency. *Struct Multidiscip Optim* 39:115–132
- Otomori M, Yamada T, Izui K, Nishiwaki S (2014) Matlab code for a level set-based topology optimization method using a reaction diffusion equation. *Struct Multidiscip Optim* 51:1159–1172
- Paulino GH, Silva ECN, Le CH (2009) Optimal design of periodic functionally graded composites with prescribed properties. *Struct Multidiscip Optim* 38:469–489
- Rodrigues H, Guedes JM, Bendsoe MP (2002) Hierarchical optimization of material and structure. *Struct Multidiscip Optim* 24:1–10
- Rozvany GIN, Bendsøe MP, Kirsch U (1995) Layout optimization of structures. *Appl Mech Rev* 48:41–119
- Sethian JA, Wiegmann A (2000) Structural boundary design via level set and immersed Interface methods. *J Comput Phys* 163:489–528
- Sigmund O (1994) Materials with prescribed constitutive parameters: an inverse homogenization problem. *Int J Solids Struct* 31:2313–2329
- Sigmund O (2001) A 99 line topology optimization code written in Matlab. *Struct Multidiscip Optim* 21:120–127
- Sigmund O (2007) Morphology-based black and white filters for topology optimization. *Struct Multidiscip Optim* 33:401–424
- Sivapuram R, Dunning PD, Kim HA (2016) Simultaneous material and structural optimization by multiscale topology optimization. *Struct Multidiscip Optim* 54:1267–1281
- Suresh K (2010) A 199-line Matlab code for Pareto-optimal tracing in topology optimization. *Struct Multidiscip Optim* 42:665–679

- Takezawa A, Nishiwaki S, Kitamura M (2010) Shape and topology optimization based on the phase field method and sensitivity analysis. *J C3comput Phys* 229:2697–2718
- Talischi C, Paulino GH, Pereira A, Menezes IFM (2012a) PolyTop: a Matlab implementation of a general topology optimization framework using unstructured polygonal finite element meshes. *Struct Multidiscip Optim* 45:329–357
- Talischi C, Paulino GH, Pereira A, Menezes IFM (2012b) PolyMesher: a general-purpose mesh generator for polygonal elements written in Matlab. *Struct Multidiscip Optim* 45:309–328
- Theocaris PS, Stavroulaki GE (1999) Optimal material design in composites: an iterative approach based on homogenized cells. *Comput Methods Appl Mech Eng* 169:31–42
- Vicente WM, Zuo ZH, Pavanello R et al (2016) Concurrent topology optimization for minimizing frequency responses of two-level hierarchical structures. *Comput Methods Appl Mech Eng* 301:116–136
- Wang C, Zhu JH, Zhang WH et al (2018a) Concurrent topology optimization design of structures and non-uniform parameterized lattice microstructures. *Struct Multidiscip Optim* 58:35–50
- Wang MY, Wang X, Guo D (2003) A level set method for structural topology optimization. *Comput Methods Appl Mech Eng* 192: 227–246
- Wang S, Wang MY (2006) Radial basis functions and level set method for structural topology optimization. *Int J Numer Methods Eng* 65: 2060–2090
- Wang Y, Chen F, Wang MY (2017a) Concurrent design with connectable graded microstructures. *Comput Methods Appl Mech Eng* 317:84–101
- Wang Y, Wang MY, Chen F (2016) Structure-material integrated design by level sets. *Struct Multidiscip Optim* 54:1145–1156
- Wang Y, Xu H, Pasini D (2017b) Multiscale isogeometric topology optimization for lattice materials. *Comput Methods Appl Mech Eng* 316:568–585
- Wang Y, Zhang L, Daynes S et al (2018b) Design of graded lattice structure with optimized mesostructures for additive manufacturing. *Mater Des* 142:114–123
- Xia L, Breitkopf P (2014) Concurrent topology optimization design of material and structure within FE2 nonlinear multiscale analysis framework. *Comput Methods Appl Mech Eng* 278:524–542
- Xia L, Breitkopf P (2015) Design of materials using topology optimization and energy-based homogenization approach in Matlab. *Struct Multidiscip Optim* 52:1229–1241
- Xia L, Breitkopf P (2017) Recent advances on topology optimization of multiscale nonlinear structures. *Arch Comput Methods Eng* 24:227–249
- Xie YM, Steven GP (1993) A simple evolutionary procedure for structural optimization. *Comput Struct* 49:885–969
- Xu B, Huang X, Zhou SW, Xie YM (2016) Concurrent topological design of composite thermoelastic macrostructure and microstructure with multi-phase material for maximum stiffness. *Compos Struct* 150:84–102
- Xu L, Cheng G (2018) Two-scale concurrent topology optimization with multiple micro materials based on principal stress orientation. *Struct Multidiscip Optim* 57:2093–2107
- Xu Y, You T, Du C (2015a) An integrated micromechanical model and BP neural network for predicting elastic modulus of 3-D multi-phase and multi-layer braided composite. *Compos Struct* 122:308–315
- Xu Y, Zhang P, Lu H, Zhang W (2015b) Hierarchically modeling the elastic properties of 2D needled carbon/carbon composites. *Compos Struct* 133:148–156
- Yan J, Duan Z, Lund E, Wang J (2017) Concurrent multi-scale design optimization of composite frames with manufacturing constraints. *Struct Multidiscip Optim* 56:519–533
- Yan X, Huang X, Sun G, Xie YM (2015) Two-scale optimal design of structures with thermal insulation materials. *Compos Struct* 120: 358–365
- Yan X, Huang X, Zha Y, Xie YM (2014) Concurrent topology optimization of structures and their composite microstructures. *Comput Struct* 133:103–110
- Yu X, Zhou J, Liang H et al (2018) Mechanical metamaterials associated with stiffness, rigidity and compressibility: a brief review. *Prog Mater Sci* 94:114–173
- Zhang H, Wang Y, Kang Z (2019) Topology optimization for concurrent design of layer-wise graded lattice materials and structures. *Int J Eng Sci* 138:26–49
- Zhang W, Dai G, Wang F et al (2007) Using strain energy-based prediction of effective elastic properties in topology optimization of material microstructures. *Acta Mech Sinica* 23:77–89
- Zhang W, Sun S (2006) Scale-related topology optimization of cellular materials and structures. *Int J Numer Methods Eng* 68:993–1011
- Zheng J, Luo Z, Jiang C, Gao J (2019) Robust topology optimization for concurrent design of dynamic structures under hybrid uncertainties. *Mech Syst Signal Process* 120:540–559
- Zheng X, Lee H, Weisgraber TH et al (2014) Ultralight, ultrastiff mechanical metamaterials. *Science* 344(80):1373–1377
- Zhou M, Rozvany GIN (1991) The COC algorithm, part II: topological, geometrical and generalized shape optimization. *Comput Methods Appl Mech Eng* 89:309–336
- Zuo ZH, Huang X, Rong JH, Xie YM (2013) Multi-scale design of composite materials and structures for maximum natural frequencies. *Mater Des* 51:1023–1034

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.