

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/286440183>

A new topology optimization approach based on Moving Morphable Components (MMC) and the ersatz material model

Article in *Structural and Multidisciplinary Optimization* · December 2015

DOI: 10.1007/s00158-015-1372-3

CITATIONS

57

READS

961

4 authors, including:



Weisheng Zhang

Dalian University of Technology

34 PUBLICATIONS **887** CITATIONS

[SEE PROFILE](#)



Jian Zhang

Delft University of Technology

5 PUBLICATIONS **145** CITATIONS

[SEE PROFILE](#)



Xu Guo

Dalian University of Technology

140 PUBLICATIONS **2,303** CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



topology optimization methods [View project](#)



Investigation about materials with different properties in tension and compression (bi-modulus materials) [View project](#)

A new topology optimization approach based on Moving Morphable Components (MMC) and the ersatz material model

Weisheng Zhang¹ · Jie Yuan¹ · Jian Zhang¹ · Xu Guo¹

Received: 10 August 2015 / Revised: 19 October 2015 / Accepted: 20 October 2015 / Published online: 9 December 2015
© Springer-Verlag Berlin Heidelberg 2015

Abstract This paper presents a new topology optimization approach based on the so-called Moving Morphable Components (MMC) solution framework. The proposed method improves several weaknesses of the previous approach (e.g., Guo et al. in *J Appl Mech* 81:081009, 2014a) in the sense that it can not only allow for components with variable thicknesses but also enhance the numerical solution efficiency substantially. This is achieved by constructing the topological description functions of the components appropriately, and utilizing the ersatz material model through projecting the topological description functions of the components. Numerical examples demonstrate the effectiveness of the proposed approach. In order to help readers understand the essential features of this approach, a 188 line Matlab implementation of this approach is also provided.

Keywords Topology optimization · Moving Morphable components · Design sensitivity

1 Introduction

Since the pioneering work of Bendsoe and Kikuchi (1988), topology optimization which aims at finding appropriate material distribution in a prescribed domain in order to get

optimized structural performances, has received considerable research attention. In the literature, numerous topology optimization approaches such as SIMP (Solid Isotropic Material with Penalization) approach (Bendsoe 1989; Zhou and Rozvany 1991), level set approach (Wang et al. 2003; Allaire et al. 2004) and evolutionary approach (Xie and Steven 1993) have been proposed. These approaches have been applied successfully to solve a wide range of topological design problems where structural, acoustics or optics performances are considered (Eschenauer and Olhoff 2001; Guo and Cheng 2010; Sigmund and Maute 2013).

Most of the existing approaches actually do topology optimization in an implicit way. This means that in these approaches the optimal structural topology is identified either from a black-and-white pixel image (in SIMP approach) or from the level set of a Topology Description Function (TDF) defined in a prescribed design domain (in level set approach). Possible problems associated with the implicit methods can be summarized as follows. Firstly, it is difficult to give a precise control of the structural feature sizes, which is very important from manufacturing point of view, under the implicit topology optimization framework (Petersson and Sigmund 1998; Poulsen 2003; Guest et al. 2004, 2011; Guest 2009a, b, 2015; Chen et al. 2008; Luo et al. 2008; Sigmund 2009; Wang et al. 2011; Sigmund and Maute 2013; Ha and Guest 2014; Zhou et al. 2015). This is because there is no explicit geometry information embedded in the implicit optimization model and special techniques must be developed to achieve a complete length scale control (Guo et al. 2014b; Zhang et al. 2014, 2015b; Michailidis 2014; Xia and Shi 2015). Furthermore, it is also not easy to establish a direct link between the optimization models and computer-aided-design (CAD) modeling system under the implicit framework since structural geometries are represented in totally different ways in these two settings. Secondly, the number of design variables

✉ Xu Guo
guoxu@dlut.edu.cn

¹ State Key Laboratory of Structural Analysis for Industrial Equipment, Department of Engineering Mechanics, Dalian University of Technology, Dalian 116023, People's Republic of China

involved in implicit topology optimization approaches is relatively large especially for three dimensional problems. The higher the resolution, the more design variables. Thirdly, in implicit approaches, analysis model and optimization model are always strongly coupled. Sometime this may lead to severe numerical problems (e.g., checkerboard pattern, spurious local vibration/buckling modes) which may prevent the optimization algorithms from converging to meaningful solutions especially when topology optimization problems in multi-physics settings are considered.

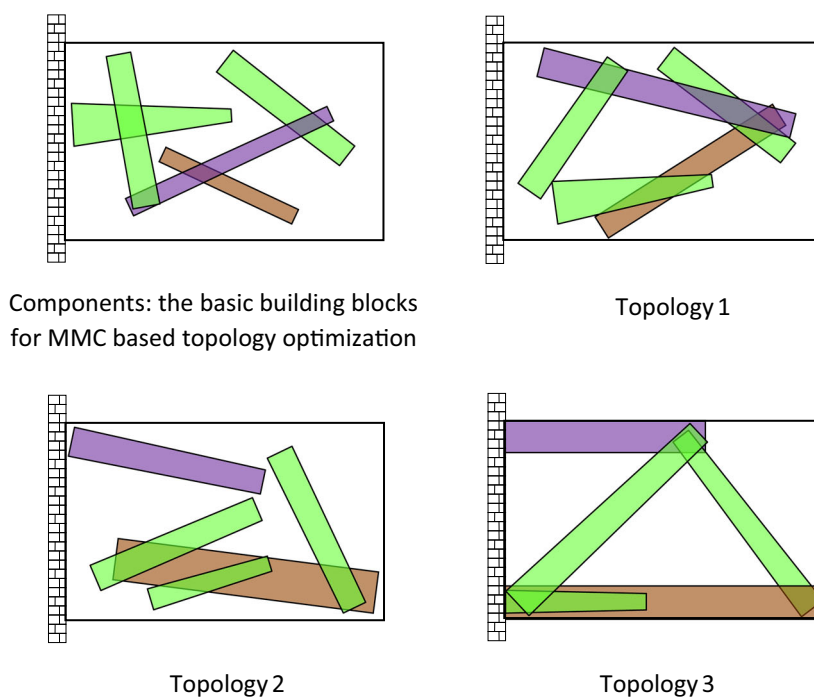
With the aim of doing topology optimization in a more explicit and geometrical way, a so-called moving morphable components based topology optimization framework, which is quite different from the existing ones, is established in (Guo et al. 2014a). The distinctive feature of this approach is that a set of morphable components are used as building blocks of topology optimization and the optimal structural topologies are found by optimizing the shapes, lengths, thicknesses, orientations and layout (connectivity) of these components. Figure 1 illustrates the basic idea of this approach schematically. Recently, the same idea has also been adopted in (Norato et al. 2015) based on the SIMP framework for topology optimization of continuum structures made of discrete elements. In this work, a new type of structural components was also introduced.

In (Guo et al. 2014a), XFEM method was employed for finite element analysis. To this end, the elements neighboring the structural boundary must be subdivided based on the nodal values of the TDFs associated with each component.

Although more accurate analysis results can be obtained by this approach, extra computational efforts are inevitably involved compared with the ersatz material model based treatment, whose effectiveness and efficiency has been demonstrated by many works especially when global objective functions (e.g., structural compliance) are concerned. Furthermore, in (Guo et al. 2014a) only components with uniform thicknesses are used as building blocks of optimization. This is obvious far from being satisfactory from geometry modeling point of view. Therefore it is highly desirable to improve the current MMC-based topology optimization framework by developing more efficient solution approaches with more flexible geometry modeling capabilities.

Another motivation of the present work comes from the following consideration. As pointed out in (Sigmund and Maute 2013), it is imperative to provide clear and well-written research code to the community when a new topology optimization approach is proposed. The provision of the codes may help readers grasp the essential features of the suggested approaches, compare them with other approaches or even rectify their deficiencies. Sigmund (2001) first presented a compact Matlab implementation of a topology optimization code for compliance minimization of continuum structures in 2001. This work inspired a series of subsequent code open work and plays a very important role in promoting development of topology optimization method. A more refined and faster version of this code is proposed by Andreassen et al. 2011. Allaire and Olivier (2006) demonstrated how to carry out efficient shape optimization using the free finite element

Fig. 1 The basic idea of the MMC- based topology optimization approach



software FreeFem++. Suresh (2010) published a 199 line Matlab code to generate Pareto-optimal topologies by exploiting the topological sensitivity information. Challis (2010) showed that a discrete level set topology optimization can be implemented in a 129 line Matlab code. Recently, Liu and Tovar (2014) presented a 169 line Matlab code for solving three dimensional topology optimization problems. Otomori et al. (2015) described a simple Matlab implementation for a level set based topology optimization method in which the level set function is updated using a reaction diffusion equation.

Based on the above considerations, in the present paper, a new topology optimization approach based on the Moving Morphable Components (MMC) solution framework is proposed. The proposed method improves several weaknesses of the previous approach (e.g., Guo et al. 2014a) in the sense that it can not only allow for components with variable thicknesses but also enhance the numerical solution efficiency substantially. This is achieved by constructing the topological description functions of the components appropriately, and utilizing the ersatz material model through projecting the topological description functions of the components. Furthermore, a Matlab code which implements the proposed approach is also presented and disseminated. For the sake of simplicity, only two dimensional problems are considered in the present work.

The rest of the paper is organized as follows. In Section 2, the preliminary ingredients of the proposed moving morphable components (MMC) based topology optimization approach are described briefly. Section 3 discusses its numerical implementation aspects. Section 4 explains the corresponding Matlab code in detail. Several numerical examples are provided in Section 5 to illustrate the effectiveness of the proposed approach. Finally, some concluding remarks are given in Section 6. The Matlab source code is given in Appendix.

2 Problem formulation

2.1 Structural shape and topology description

As shown in (Guo et al. 2014a), in the MMC based approach, structural topology description can be achieved in the following way:

$$\begin{cases} \phi^s(\mathbf{x}) > 0, & \text{if } \mathbf{x} \in \Omega^s, \\ \phi^s(\mathbf{x}) = 0, & \text{if } \mathbf{x} \in \partial\Omega^s, \\ \phi^s(\mathbf{x}) < 0, & \text{if } \mathbf{x} \in D \setminus \Omega^s. \end{cases} \quad (1)$$

In (1), D represents a prescribed design domain and $\Omega^s \subset D$ denotes a subset of D occupied by n components made of solid material. We also have $\phi^s(\mathbf{x}) = \max(\phi_1, \dots, \phi_n)$ with

$\phi_i = \phi_i(\mathbf{x})$, $i = 1, \dots, n$, denoting the topology description function (TDF) of the region occupied by the i -th component (i.e., Ω_i), that is,

$$\begin{cases} \phi_i(\mathbf{x}) > 0, & \text{if } \mathbf{x} \in \Omega_i, \\ \phi_i(\mathbf{x}) = 0, & \text{if } \mathbf{x} \in \partial\Omega_i, \\ \phi_i(\mathbf{x}) < 0, & \text{if } \mathbf{x} \in D \setminus \Omega_i. \end{cases} \quad (2)$$

Obviously, $\Omega^s = \cup_{i=1}^n \Omega_i$. We refer the readers to Fig. 2 for a schematic illustration of the above geometry representation.

In (Guo et al. 2014a) only components with uniform thicknesses are used as building blocks of optimization. In order to allow for more general case where the components can have variable thicknesses, in the present paper, we propose to use the following TDF to represent the geometry of the i -th component explicitly:

$$\phi_i(x, y) = \left(\frac{x'}{L_i}\right)^p + \left(\frac{y'}{f(x')}\right)^p - 1 \quad (3)$$

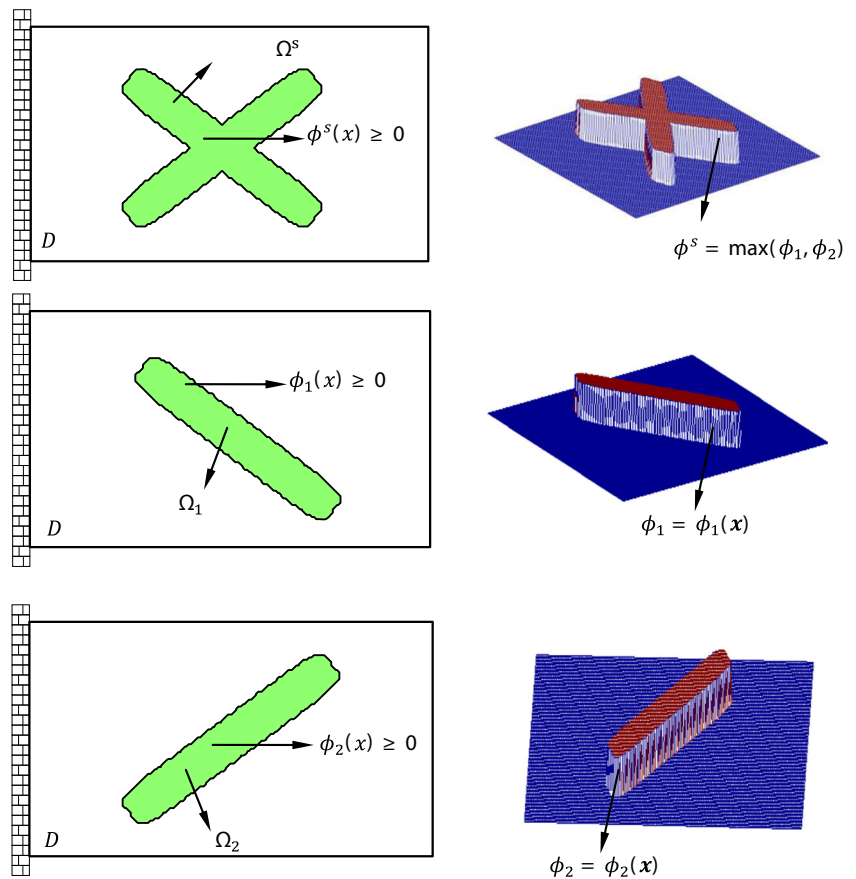
with

$$\begin{Bmatrix} x' \\ y' \end{Bmatrix} = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix} \begin{Bmatrix} x - x_{0i} \\ y - y_{0i} \end{Bmatrix} \quad (4)$$

and p is a relatively large even integer number (we take $p=6$ in the present study). In (3 and 4), (x_{0i}, y_{0i}) denotes the coordinate of the center of the component, L_i denotes the half length of the component and θ_i (measured from the horizontal axis anti-clockwisely) is the inclined angle of the component, respectively. These parameters described the shape of the components *explicitly*. Compared to the TDF adopted in (Guo et al. 2014a), the present TDF can represent the shape of a component with variable thickness whose profile is controlled by $f(x')$. This treatment will enhance the geometry modeling capability of the MMC approach substantially. Figure 3 depicts the shapes of the corresponding components when $f(x')$ takes different forms.

It is worth noting that although TDF has been used to represent the geometry of a component in the MMC based approach, the MMC based approach is quite different from the conventional level set approaches. The difference consists in the fact that in the proposed MMC based method, it is possible to give an explicit description of the boundary and geometry features (e.g., length and thickness) of a component. This, however, cannot be achieved in conventional level set approaches where free-form or radial basis level set functions are used to represent the boundary of a structure. Here, the so-called explicit boundary description means that a unique explicit relationship can be established locally between x_b and y_b , where (x_b, y_b) denoting the coordinates of a point on the structural boundary. In other words, y_b can be expressed as an explicit function of x_b , i.e., $y_b = y_b(x_b; \mathbf{D}^i)$, where \mathbf{D}^i represents the vector of design variables associated with the i -th

Fig. 2 The representation of structural topology through the level set functions of each component



component on which (x_b, y_b) is located. It is also worthwhile to point out that for components with more complex shapes, the corresponding TDFs containing explicit geometrical information can also be constructed in a systematic way (Guo et al. 2015).

In summary, under the above geometry representation scheme, the layout (i.e., shape and topology) of a structure can be solely determined by a design vector $\mathbf{D} = ((\mathbf{D}^1)^\top, \dots, (\mathbf{D}^i)^\top, \dots, (\mathbf{D}^n)^\top)^\top$. Here $\mathbf{D}^i = (x_{0i}, y_{0i}, L_i, \theta_i, \mathbf{d}_i^\top)^\top$ and the symbol \mathbf{d}_i denotes the vector of parameters associated with $f(x)$ (e.g., t_i^1 , t_i^2 and t_i^3 in Fig. 3).

2.2 Problem formulation

As shown in (Guo et al. 2014a), topology optimization problem under MMC based framework can be formulated as follows:

$$\begin{aligned} &\text{Find } \mathbf{D} = ((\mathbf{D}^1)^\top, \dots, (\mathbf{D}^i)^\top, \dots, (\mathbf{D}^n)^\top)^\top \\ &\text{Minimize } I = I(\mathbf{D}) \end{aligned}$$

s.t.

$$\begin{aligned} &g_j(\mathbf{D}) \leq 0, \quad j = 1, \dots, m, \\ &\mathbf{D} \in \mathcal{U}_D, \end{aligned} \quad (5)$$

where \mathbf{D}^i , $i = 1, \dots, n$ is the vector of the design variables associated with the i -th component, respectively. In (5), g_j , $j = 1, \dots, m$ are the constraint functions/functional and \mathcal{U}_D is the admissible set that \mathbf{D} belongs to.

If compliance minimization under available volume constraint is considered, the corresponding problem formulation can be written as

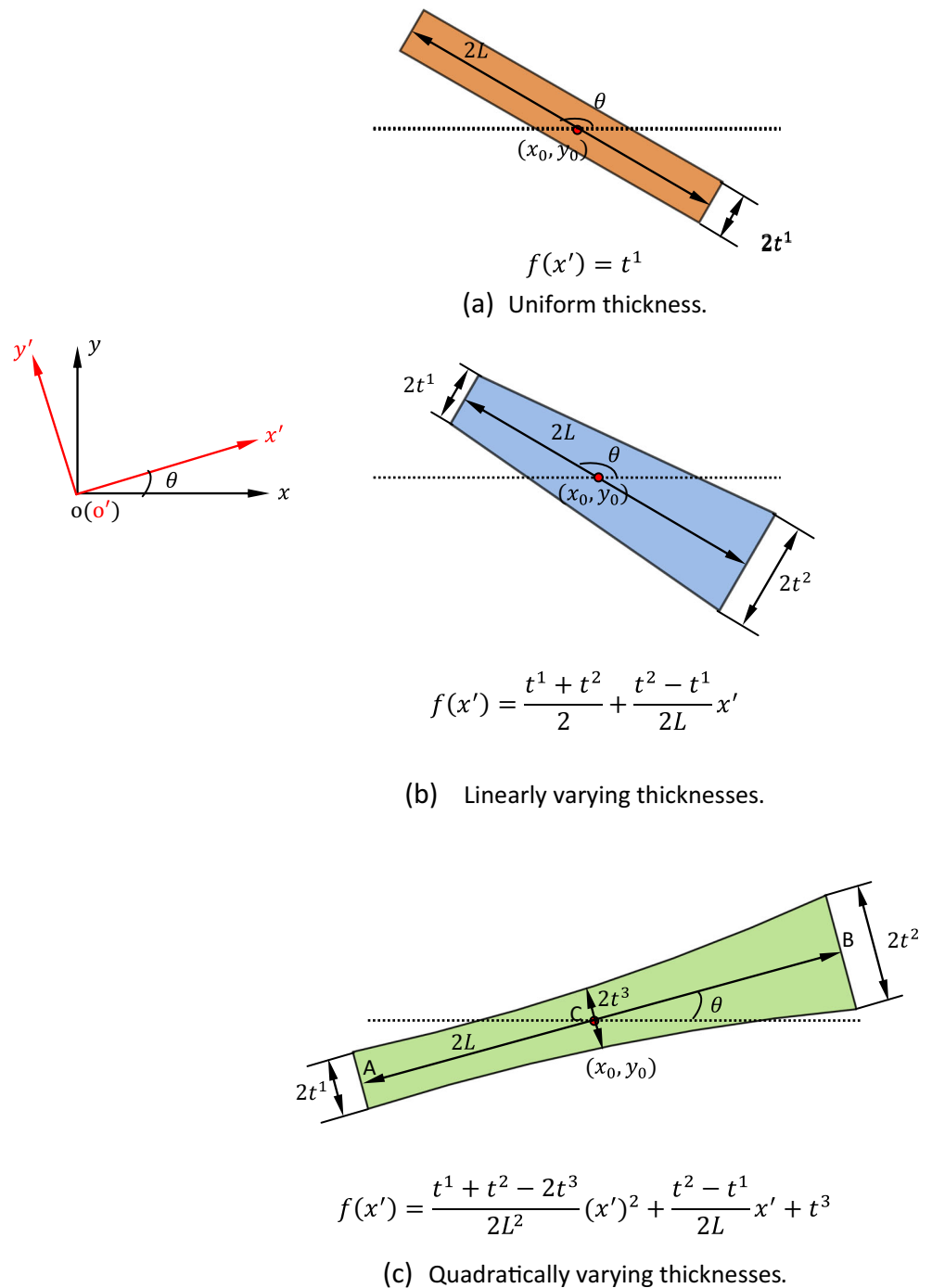
$$\begin{aligned} &\text{Find } \mathbf{D} = ((\mathbf{D}^1)^\top, \dots, (\mathbf{D}^i)^\top, \dots, (\mathbf{D}^n)^\top)^\top, \mathbf{u}(\mathbf{x}) \\ &\text{Minimize } C = \sum_{i=1}^n \int_{\Omega^i \setminus (\cup_{1 \leq j < i} (\Omega^j \cap \Omega^i))} \mathbf{f}^i \cdot \mathbf{u} dV + \int_{\Gamma_t} \mathbf{t} \cdot \mathbf{u} dS \end{aligned}$$

s.t.

$$\begin{aligned} &\sum_{i=1}^n \int_{\Omega^i \setminus (\cup_{1 \leq j < i} (\Omega^j \cap \Omega^i))} \mathbb{E}^i : \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) dV = \\ &\sum_{i=1}^n \int_{\Omega^i \setminus (\cup_{1 \leq j < i} (\Omega^j \cap \Omega^i))} \mathbf{f}^i \cdot \mathbf{v} dV + \int_{\Gamma_t} \mathbf{t} \cdot \mathbf{v} dS, \quad \forall \mathbf{v} \in \mathcal{U}_{ad}, \\ &V(\mathbf{D}) \leq \bar{V}, \\ &\mathbf{D} \in \mathcal{U}_D, \\ &\mathbf{u} = \bar{\mathbf{u}}, \text{ on } \Gamma_u. \end{aligned} \quad (6)$$

In (6), \mathbf{f}^i and \mathbf{t} denote the body force density in Ω^i , $i = 1, \dots, n$ and the surface traction on Neumann boundary Γ_t , respectively. $\bar{\mathbf{u}}$ is the prescribed displacement on Dirichlet boundary

Fig. 3 Geometry description of a structural component



Γ_u . In addition, \mathbf{u} and \mathbf{v} are the displacement field and the corresponding test function defined on $\Omega = \cup_{i=1}^n \Omega^i$ with $\mathcal{U}_{ad} = \{\mathbf{v} | \mathbf{v} \in \mathbf{H}^1(\mathbf{D}), \mathbf{v} = \mathbf{0} \text{ on } \Gamma_u\}$. The symbol $\boldsymbol{\varepsilon}$ represents the second order linear strain tensor. $\mathbb{E}^i = E^i / (1 + \nu^i) [\mathbb{I} + \nu^i / (1 - 2\nu^i) \boldsymbol{\delta} \otimes \boldsymbol{\delta}]$ (\mathbb{I} and $\boldsymbol{\delta}$ denote the fourth and the second order identity tensor, respectively) is the fourth order isotropic elasticity tensor of the material constituting the i -th component. In the expression of \mathbb{E}^i , the symbols

E^i and ν^i denote the corresponding Young's modulus and Poisson's ratio, respectively. In (6), it has been assumed that $\mathbb{E}(\mathbf{x}) = \mathbb{E}^i$ and $\mathbf{f} = \mathbf{f}^i$ if $\mathbf{x} \in \Omega^i \setminus (\cup_{1 \leq j < i} (\Omega^j \cap \Omega^i))$, $i = 1, \dots, n$, respectively. The symbol \bar{V} denotes the upper bound of the available volume of solid material. In the present work, it is assumed that $E^i = \dots = E^n = E$ and $\nu^1 = \dots = \nu^n = \nu$, respectively. For the sake of simplicity, we set $\bar{\mathbf{u}} = \mathbf{0}$ in the following discussions.

When only single-phase material is considered, (6) can be rewritten in terms of $\phi^s(\mathbf{x})$ as follows:

$$\begin{aligned} & \text{Find } \mathbf{D} = ((\mathbf{D}^1)^\top, \dots, (\mathbf{D}^i)^\top, \dots, (\mathbf{D}^n)^\top)^\top, \mathbf{u}(\mathbf{x}) \\ & \text{Minimize } C = \int_D H(\phi^s(\mathbf{x}; \mathbf{D})) \mathbf{f} \cdot \mathbf{u} dV + \int_{\Gamma_t} \mathbf{t} \cdot \mathbf{u} dS \\ & \text{s.t.} \\ & \int_D H(\phi^s(\mathbf{x}; \mathbf{D})) \mathbb{E} : \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) dV = \int_D H(\phi^s(\mathbf{x}; \mathbf{D})) \mathbf{f} \cdot \mathbf{v} dV \\ & + \int_{\Gamma_t} \mathbf{t} \cdot \mathbf{v} dS, \forall \mathbf{v} \in \mathcal{U}_{ad}, \\ & \int_D H(\phi^s(\mathbf{x}; \mathbf{D})) dV \leq \bar{V}, \\ & \mathbf{D} \in \mathcal{U}_D, \\ & \mathbf{u} = \bar{\mathbf{u}}, \text{ on } \Gamma_u, \end{aligned} \quad (7)$$

where $H = H(x)$ is the Heaviside function.

Since $H = H(x) = 0$, if $x \leq 0$ and $H = H(x) = 1$, otherwise, (7) is obviously equivalent to

$$\begin{aligned} & \text{Find } \mathbf{D} = ((\mathbf{D}^1)^\top, \dots, (\mathbf{D}^i)^\top, \dots, (\mathbf{D}^n)^\top)^\top, \mathbf{u}(\mathbf{x}) \\ & \text{Minimize } C = \int_D H(\phi^s(\mathbf{x}; \mathbf{D})) \mathbf{f} \cdot \mathbf{u} dV + \int_{\Gamma_t} \mathbf{t} \cdot \mathbf{u} dS \\ & \text{s.t.} \\ & \int_D (H(\phi^s(\mathbf{x}; \mathbf{D})))^q \mathbb{E} : \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) dV = \int_D H(\phi^s(\mathbf{x}; \mathbf{D})) \mathbf{f} \cdot \mathbf{v} dV \\ & + \int_{\Gamma_t} \mathbf{t} \cdot \mathbf{v} dS, \forall \mathbf{v} \in \mathcal{U}_{ad}, \\ & \int_D H(\phi^s(\mathbf{x}; \mathbf{D})) dV \leq \bar{V}, \\ & \mathbf{D} \in \mathcal{U}_D, \\ & \mathbf{u} = \bar{\mathbf{u}}, \text{ on } \Gamma_u, \end{aligned} \quad (8)$$

where $q > 1$ is an integer.

2.3 Distinctive features of the MMC based approach

Compared with the existing topology optimization approaches, the afore-mentioned MMC based approach has the following distinctive features. Firstly, unlike in traditional approaches where structural topology is represented implicitly by image pixels or nodal values of level set functions, the MMC based approach uses a set of geometry parameters to describe the structural topology explicitly. This not only reduces the number of design variables significantly (especially for three dimensional problems!) but also eliminates the image processing related issues (ensuring black-and-white design, filtering of sensitivities, suppression of boundary diffusion) in the solution process in a fundamental way. Secondly, since explicit, accurate and analytical boundary information are embedded intrinsically in the MMC based solution approaches, it may be relatively easy for MMC based approach to handle topology optimization problems involving complicated boundary layer physics (turbulence, electromagnetic wave

reflection) and boundary conditions. Thirdly, the MMC based approach has the capability to integrate the size, shape, topology and layout optimization or even structural type optimization in a unified solution framework. Furthermore, since analysis model and design model are totally decoupled in MMC based approach, the modeling capabilities can be enhanced significantly in the MMC approach since different components can be modeled using different appropriate elements, which cannot be achieved easily in traditional approaches. For other potential advantages of the MMC based approach, we refer the readers to (Guo et al. 2014a) for more details.

3 Numerical implementation

In this Section, some aspects of the numerical implementation of the MMC based approach with use of the ersatz material model will be described.

3.1 Finite element analysis

As in most topology optimization approaches, structured four-node bi-linear elements are used to discretize the design domain uniformly. As shown in (Guo et al. 2014a; Zhang et al. 2015a), since the boundaries of the components can be described accurately and explicitly in the MMC solution framework, finite element method (FEM) analysis with high accuracy can be achieved by resorting to the X-FEM approaches or body-fitted adaptive mesh techniques. In the present work, however, in order to enhance the computational efficiency, the ersatz material model is adopted for FEM analysis. With use of the ersatz material model, as shown in (Guo et al. 2005), once the values of TDF at four nodes of an element are known, the Young's modulus of this element can be interpolated as

$$E^e = \frac{E \left(\sum_{i=1}^4 (H(\phi_i^e))^q \right)}{4}, \quad (9)$$

according to (8). In (9), $H = H(x)$ is the Heaviside function and ϕ_i^e , $i = 1, \dots, 4$ are the values of the TDF function of the whole structure (i.e., $\phi^s(\mathbf{x})$) at four nodes of element e . For numerical implementation purpose, as a common practice in the literature, $H(x)$ is often replaced by its regularized version $H_\epsilon(x)$. In the present work, the form of $H_\epsilon(x)$ is taken as

$$H_\epsilon(x) = \begin{cases} 1, & \text{if } x > \epsilon, \\ \frac{3(1-\alpha)}{4} \left(\frac{x}{\epsilon} - \frac{x^3}{3\epsilon^3} \right) + \frac{(1+\alpha)}{2}, & \text{if } -\epsilon \leq x \leq \epsilon, \\ \alpha, & \text{otherwise,} \end{cases} \quad (10)$$

where ϵ is a parameter that controls magnitude of regularization and α is a small positive number to ensure the nonsingular

of the global stiffness matrix. In addition, in the present study, we take $q=2$ in (9).

3.2 Design sensitivity

Only design sensitivity analysis for compliance minimization problem is discussed here. Extensions to allow for more general objective/constraint functions can be achieved easily by resorting to adjoint sensitivity analysis.

If the concerned objective function is the structural compliance, it is well known that its sensitivity with respect to an arbitrary geometry parameter a can be written as:

$$\frac{\partial C}{\partial a} = -\mathbf{u}^T \frac{\partial \mathbf{K}}{\partial a} \mathbf{u} = -\mathbf{u}^T \left(\frac{E}{4} \left(\sum_{e=1}^{NE} \sum_{i=1}^4 q(H(\phi_i^e))^{q-1} \frac{\partial H(\phi_i^e)}{\partial a} \right) \mathbf{k}^s \right) \mathbf{u}, \quad (11)$$

where \mathbf{K} is the global stiffness matrix of the structure and \mathbf{k}^s is the element stiffness matrix corresponding to $\phi_i^e=1$, $i=1, \dots, 4$ and $E=1$. In (11), the symbol NE denotes the total number of elements in the ground structure. As shown in (Guo et al. 2014a), $\partial H(\phi_i^e)/\partial a$ can be calculated easily since $\phi^s(\mathbf{x})$ is an explicit function of a . In the present work, however, in order to enhance the universality of the code, we use the finite difference quotient of ϕ_i^e (i.e., $\Delta H(\phi_i^e)/\Delta a$) to calculate $\partial H(\phi_i^e)/\partial a$ approximately. Since $\phi^s(\mathbf{x})$ is an explicit function of a and the finite difference operation is taken only at the element level, the calculation of $\Delta H(\phi_i^e)/\Delta a$ is very fast. Numerical examples show that this treatment performs very well for the problems considered.

In addition, we also have

$$\frac{\partial V}{\partial a} = \frac{1}{4} \sum_{e=1}^{NE} \sum_{i=1}^4 \frac{\partial H(\phi_i^e)}{\partial a}. \quad (12)$$

It also worth noting that in the above derivations, the non-differentiability issue arising from the max operation when ϕ^s

is constructed by $\phi^s(\mathbf{x}) = \max(\phi_1, \dots, \phi_n)$ is neglected. Substantial numerical evidences indicate that this treatment has no influence on the optimization process.

3.3 Optimization algorithm

In the present work, the well-known MMA optimizer (Svanberg 1987) is adopted to solve the topology optimization problems formulated in (5).

4 Matlab implementation

In order to help readers understand the essential features of this approach, a Matlab code for structural compliance minimization problems based on the proposed approach is documented in Appendix. This code is corresponding to the short beam problem in Section 5. The main purpose of the code is neither to present every aspect of the MMC approach nor give a fine implementation of it. It is just intended to help readers understand the essential features of the proposed approach. Actually, the code inherits most of the program architecture, data structure, notation of symbols from the well-known 99 and 88 line code of (Sigmund 2001) and (Andreassen et al. 2011), which implements the SIMP approach. The FEM discretization, setting of the boundary and loading conditions are all the same as in Sigmund's 99 lines code. The code includes four parts that will be disseminated in detail in the following.

4.1 Main function

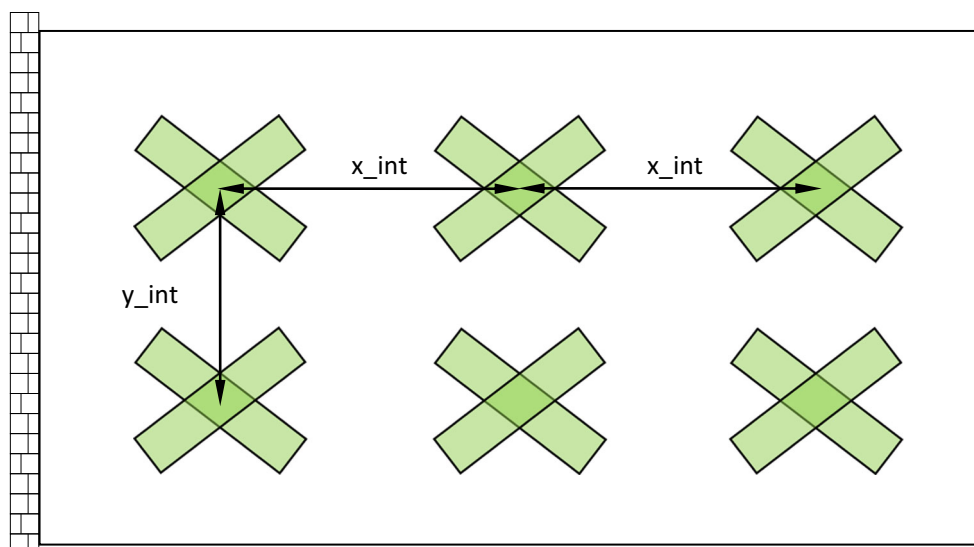
The Matlab code is evoked by the following command line:

MMC188 (DW, DH, nelx, nely, x_int, y_int, ini_val, volfrac). The meaning of every symbol in the command line can be found in Table 1. It is worth noting that in the present code, for all tested examples, the initial configurations of the components are taken the same form as shown in Fig. 4. The

Table 1 The meanings of parameters in the code

Parameter	Meaning	Parameter	Meaning
DW	Width of the design domain	x0 y0	The center coordinate of a component
DH	Height of the design domain	L	The half length of a component
nelx	Number of elements in x direction	t1	The half thickness of a component at point A
nely	Number of elements in y direction	t2	The half thickness of a component at point B
x_int	Distance of components in x direction	t3	The half thickness of a component at point C
y_int	Distance of components in y direction	st	Sine value of a component's inclined angle
ini_val	Initial configuration of components	xmin	Lower bound of design variables
volfrac	Maximum allowable volume fraction	xmax	Upper bound of design variables
E	Young's modulus	m	Number of constraints
nu	Poisson's ratio	Var_num	Number of design variables

Fig. 4 The initial configuration of the components



meanings of the symbols x_int and y_int are also illustrated schematically in the same Figure. The vector ini_val stores the values of L, t^1, t^2, t^3 and $\sin\theta$ which describe the initial configurations of the components. The readers can also try other forms of initial designs by making the corresponding modifications in the code easily.

4.2 Data initialization for finite element analysis (FEA): line 3–12

This part initializes the data for finite element analysis.

4.3 Data initialization for component geometries: line 13–26

This part initializes the data describing the component geometries. In the present code, a quadratic function is used to describe the width variation of each component for all test examples. The meanings of t^1, t^2 and t^3 are also illustrated schematically in Fig. 3. In Line 26, all initial values of the design variables (7 for each component, i.e., $(x_{0i}, y_{0i}, L_i, t_i^1, t_i^2, t_i^3, \sin\theta_i)^T$) have been set up. It is worth noting that in the present implementation, $\sin\theta_i$ but not θ_i is used as design variable for numerical stability considerations.

4.4 Data initialization for MMA: line 27–45

This part initializes the data for MMA optimizer. It is noted that in the present code, the lower bound of the length and width of each component is set to a small but nonzero value. This will not prevent the change of structural topology since in the MMC based framework, topology change can be achieved mainly through overlapping and hiding mechanisms. A

component can also be eliminated from the design domain once its length or the maximum value of t^1, t^2 and t^3 is less than the length of one mesh grid.

4.5 Setting boundary and loading conditions: line 46–51

This part sets up the boundary and loading conditions for FEA. The readers can make corresponding modifications for different problems.

4.6 Preparation the data for finite element analysis: line 52–60

This part makes a preparation for FEA. In Line 60, k^s is formed.

4.7 Main loop for optimization: line 61–149

- (1): line 61–65: Setting up parameters for the regularized Heaviside function;

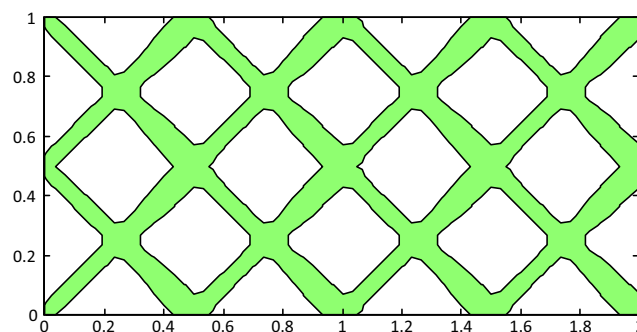


Fig. 5 The initial design for the short beam example

- (2): line 71–79: Setting the values of $\phi^s(\mathbf{x})$ at every grid node according to the current values of design variables;
- (3): line 80–82: Figure plotting;
- (4): line 84: Setting the values of $H(\mathbf{x})=H(\phi^s(\mathbf{x}))$ at every grid node according to the current value of $\phi^s(\mathbf{x})$;
- (5): line 86–112: Calculating the finite difference quotient of $H=H(\mathbf{x})$ with respect to every design variable by finite difference method;
- (6): line 113–120: Finite element analysis;
- (7): line 121–132: Calculating the sensitivities of the objective and constraint functions with respect to design variables according to (11) and (12), respectively.
- (8): line 133–149: Optimization by calling MMA optimizer. The parameters in MMA for all examples are setting as follows (in the file of mmasub.m):

```
epsimin=10^(-10);
raa0=0.01;
albefa=0.4;
asyinit=0.1;
asyincr=0.8;
asydecr=0.6;
```

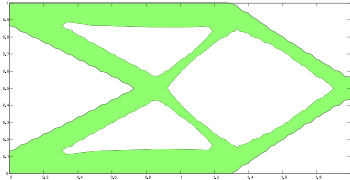
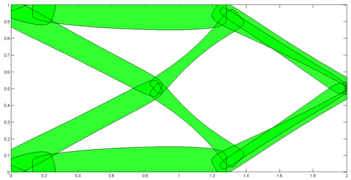
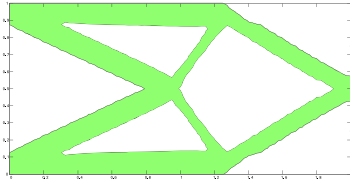
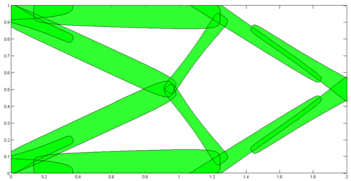
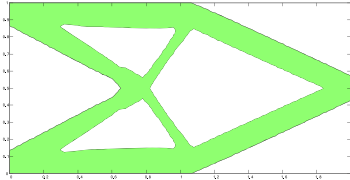
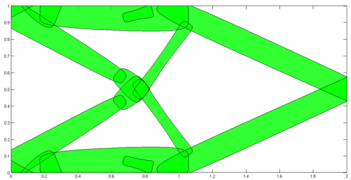
4.8 Explanation of three functions:

- (1). function [tmpPhi]=tPhi (xy, LSgridx, LSgridy, p): Forming $\phi_i=\phi_i(\mathbf{x})$ for each component on each grid node. The readers can make corresponding modifications here to allow for components with different shapes.
- (2). function H=Heaviside (phi, alpha, nelx, nely, epsilon): Calculating the value of H_ϵ on each grid node. The readers may adjust the value of ϵ (i.e., epsilon) to test of the performance of the approach for different problems.
- (3). function [KE]=BasicKe (E, nu, a, b, h): Forming \mathbf{k}^s .

5 Examples

In this section, several benchmark numerical examples are presented to demonstrate the effectiveness of the proposed approach.

Table 2 Optimal topologies of the short beam example

Mesh grids	Contour plot	Component plot
60 × 30	 $c = 76.52$	
80 × 40	 $c = 74.68$	
100 × 50	 $c = 74.66$	

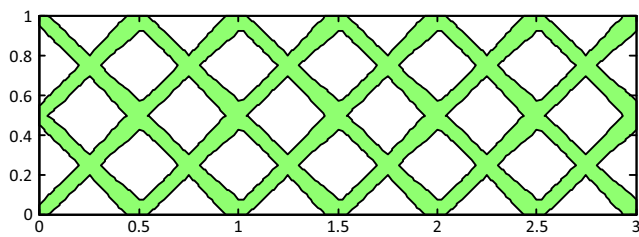


Fig. 6 The initial design for the MBB example

5.1 The short beam problem

The description of this problem can be found in (Guo et al. 2014a). The initial design is shown in Fig. 5. For this problem we set $\text{ini_val}=[0.38 \ 0.04 \ 0.06 \ 0.04 \ 0.7]$ and the design domain is discretized by a 60×30 , 80×40 and 100×50 FEM mesh, respectively. The Matlab code is evoked with use of the following call (the readers should

change the value of nelx and nely in the function for different FEM mesh settings):

MMC188 (2, 1, 80, 40, 0.5, 0.5, ini_val, 0.4)

Table 2 gives the optimal values of the objective function and the contour plots as well as the component plots of the optimized structures obtained under different meshes. It can be observed that if the number of components in the initial design is fixed, the topologies of the optimized structures are independent on the FEM mesh adopted even though no filtering is used in the present MMC based approach. This is essentially due to the fact that when the number of components is fixed, the problem in (7 and 8) is essentially an optimization problem in a finite dimension space. Under this circumstance, the considered problem is definitely well-posed since every bounded closed set (e.g., \mathcal{U}_D in (7 and 8)) in a finite dimensional space is compact. It is also

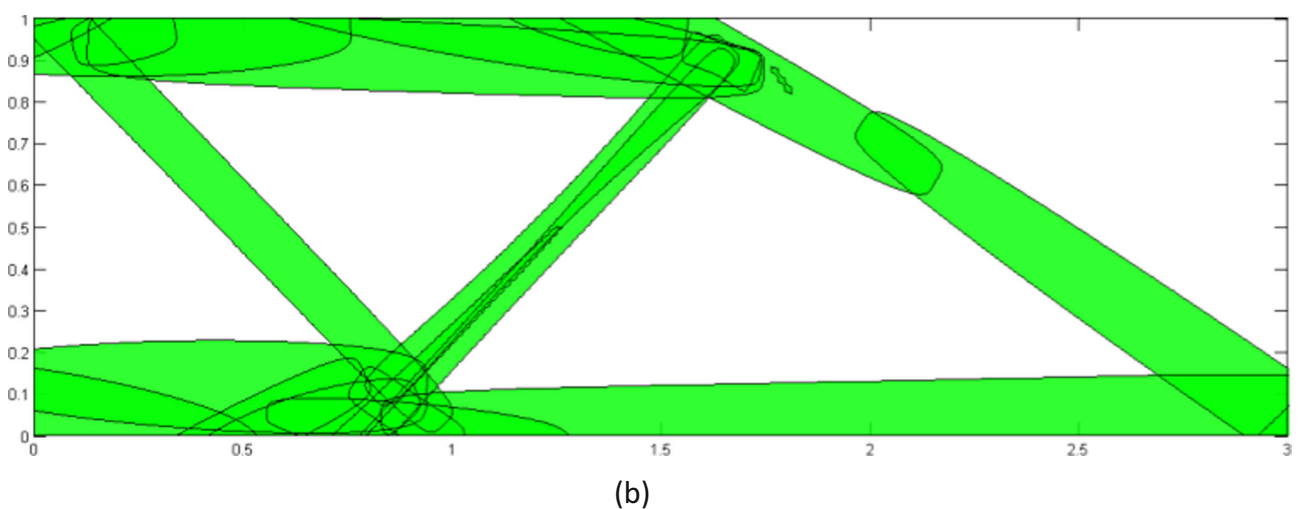
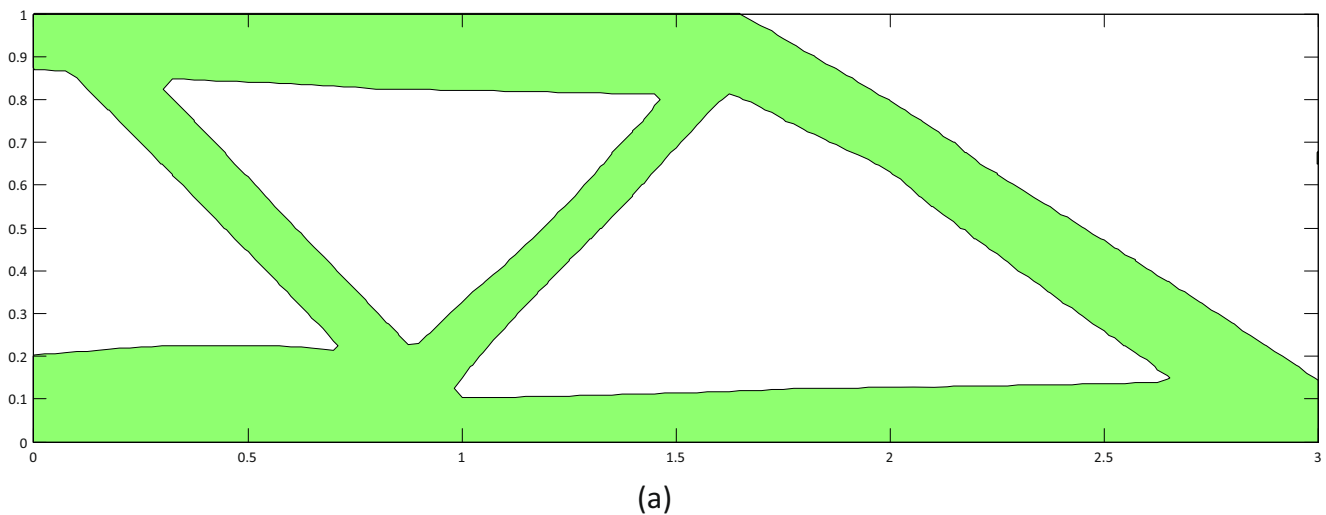


Fig. 7 Optimal topology of the MBB example: (a) the contour plot and (b) the component plot

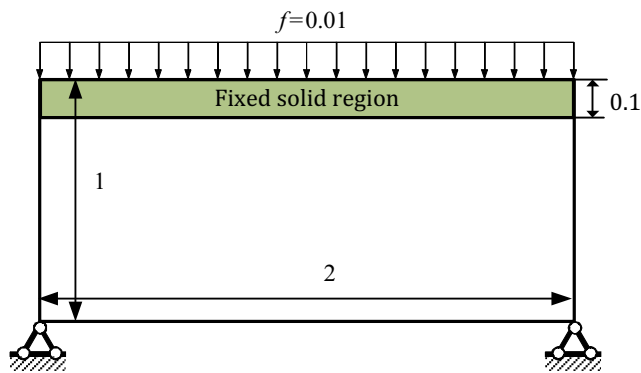


Fig. 8 The bridge example

worth noting that the somewhat non-smooth boundary in the contour plot is only due to graphical displaying reasons. Actually, the boundaries of the components are very smooth which can be seen very clearly in the component plots. In addition, the number of design variables is only $16 \times 7 = 112$ for this problem. This value is much smaller than that in conventional approaches and is independent on the FEM mesh resolution.

5.2 The MMB problem

The description of this problem can be found in (Guo et al. 2014a) and only half design domain is considered due to the symmetry property of this problem. The initial design is shown Fig. 6. For this problem, we set $\text{ini_val} = [0.38 \ 0.04 \ 0.06 \ 0.04 \ 0.7]$ and the boundary and loading conditions (lines 47 and 50 of the code) should read as:

```
fixeddofs=[ 1:2:2*(nely+1), 2*(nely+1)
*nelx+2] ;
load dof=2*(nely+1) ;
```

and the Matlab code is evoked with us of the following call:

```
MMC188 (3, 1, 120, 40, 0.5, 0.5, ini_val, 0.4)
```

Figure 7 depicts the contour plot (Fig. 7a) and component plot (Fig. 7b) of the optimized structure, respectively. It is worth noting that the optimized structure obtained by the proposed approach is pure black-and-white and there are no grey transition zones at all in the structure.

5.3 The bridge example

The third example under consideration is a bridge design problem. The loading and boundary conditions are depicted in Fig. 8. In this example, the design domain is discretized by a 80×40 FEM mesh and a rectangle zone has the width and length of 0.1×2 on the top of the domain is setting as the fixed solid region. Four intersection components are distributed in the design domain as the initial design (shown in Fig. 9). The design objective is also to minimize the mean compliance of the structure with the volume constraint such that $V \leq \bar{V} = 0.3$. It is worth noting that for this and the next compliant problem, the corresponding codes can be downloaded from the website. The contour plot and component plot of the optimized structure are depicted in Fig. 10, respectively. The Matlab code of this problem can be obtained by contacting the authors directly.

5.4 The compliant mechanism problem

The last example under consideration is the classical compliant mechanism design problem (Sigmund 2001), which is

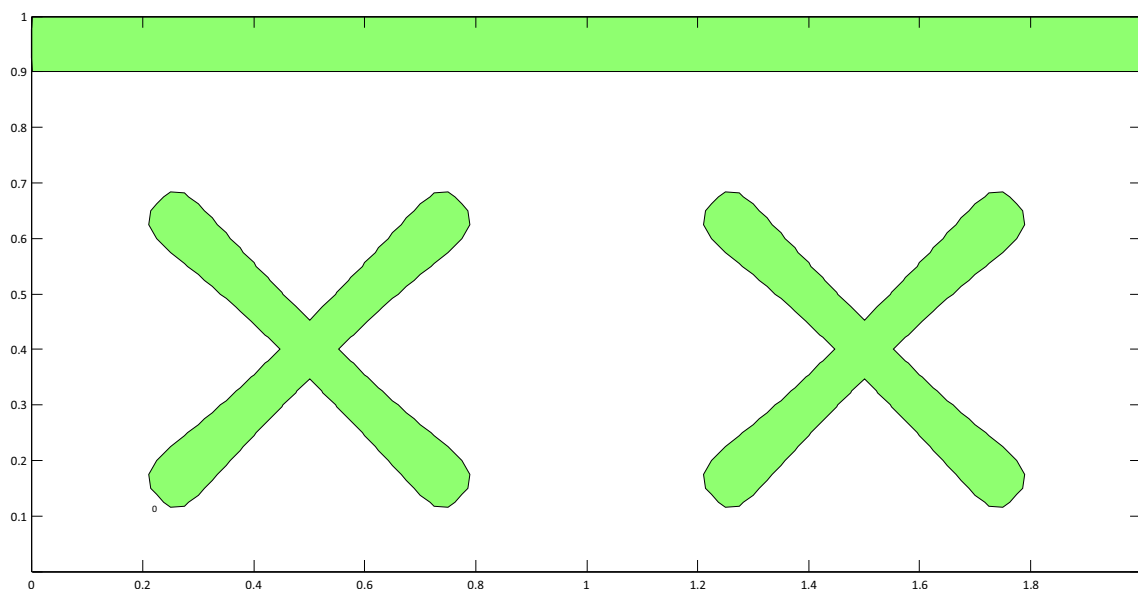


Fig. 9 The initial design for the bridge example

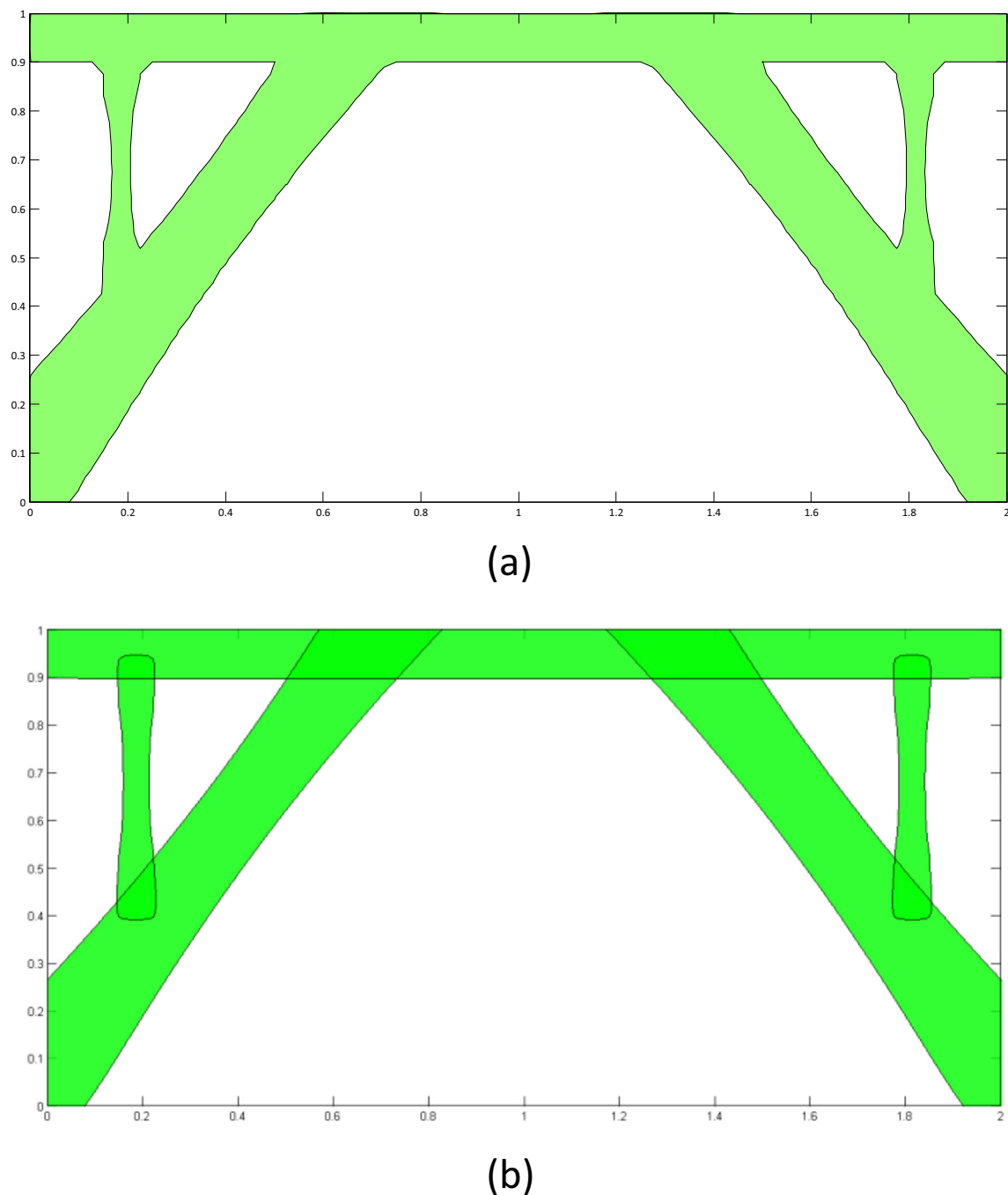


Fig. 10 Optimal topology of the bridge example: (a) the contour plot and (b) the component plot

obvious a non-self adjoint one. The corresponding data such as design domain, boundary conditions and geometry data, are depicted in Fig. 11. The actuation force is $F_{in}=1$ and the elastic constants of the springs at the input/output points are $k_{in}=k_{out}=0.1$, respectively. Since the problem under consideration is symmetric in nature, only half of the structure is optimized and discretized by a 80×40 FEM mesh. The initial design is shown Fig. 12.

For this problem, we set $ini_val=[0.38 \ 0.04 \ 0.06 \ 0.04 \ 0.7]$. The Matlab code of this problem can be obtained by making

some modifications on the code corresponding to the short beam problem presented in Appendix as follows:

- (1) Replacing line 47-line 51 in the original code by the following content:

```
fixeddofs=union([ 2*(nely+1) : 2*(nely+
1):2*(nely+1)*(nelx+1)],[ 1:6] );
alldofs=1:2*(nely+1)*(nelx+1);
freedofs=setdiff(alldofs,fixeddofs);
```

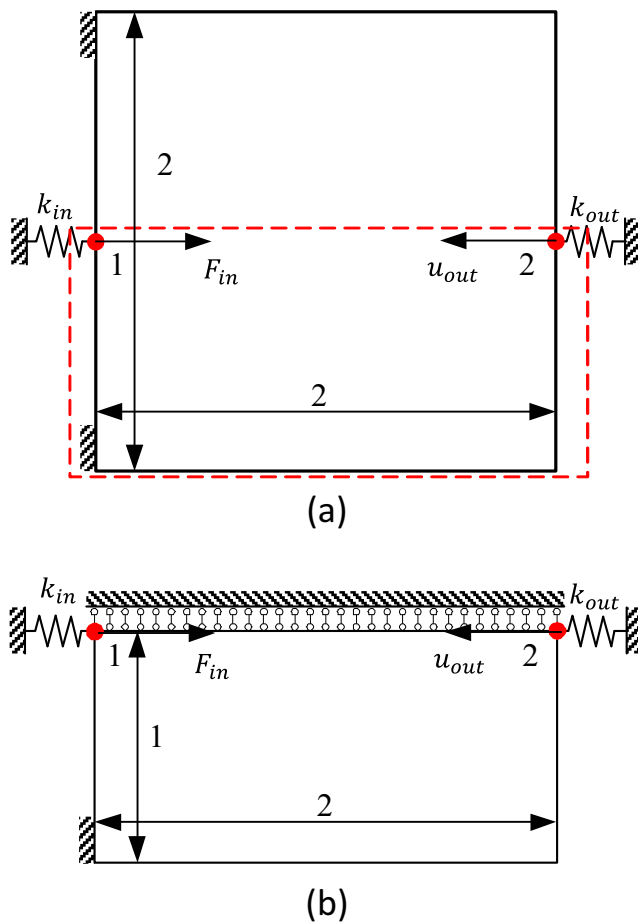


Fig. 11 The compliant mechanism example

```

InputForceID=[ 2*(nely+1)-1];
OutputForceID=[ 2*(nelx+1)*(nely+1)-1];
InputUnitForce=sparse(InputForceID,1,
1,2*(nely+1)*(nelx+1),1);
OutputUnitForce=sparse(OutputForceID,
1,-1,2*(nely+1)*(nelx+1),1);

```

- (2) Replacing line 114–line 122 in the original code by the following content:

```

denk=sum(H(ElNodesID).^2,2)/4;
den=sum(H(ElNodesID),2)/4;
A1=sum(den)*EW*EH;
U=zeros(2*(nely+1)*(nelx+1),2);
sK=KE(:)*denk(:)';
K=sparse(iK(:),jK(:),sK(:)); K=(K+K')/2;
K(InputForceID,InputForceID)=K
(InputForceID,InputForceID)+0.1;
K(OutputForceID,OutputForceID)=K
(OutputForceID,OutputForceID)+0.1;

```

```

U(freedofs,:)=K(freedofs,freedofs)\
[InputUnitForce(freedofs,:),
OutputUnitForce(freedofs,:)];
U1=U(:,1);
U2=U(:,2);
U1(fixeddofs,:)=0;
U2(fixeddofs,:)=0;
%Energy of nodes
energy=-sum((U1(edofMat)*KE).*U2
(edofMat),2);

```

- (3) Replacing line 133 in the original code by the following sentence

```
Comp=-sum(energy.*denk(:));
```

- The Matlab code is evoked with us of the following call:

```
MMC188(2,1,80,40,0.5,0.5,ini_val,0.3)
```

The optimized structure of this problem is shown in Fig. 13 with the contour plot and the component plot, respectively. It can be observed that some of the unnecessary components have been overlapped by other components. This is actually an effective mechanism to realize topology changes in the MMC based approach. It is also worth noting that there is no hinge like connection in the final optimized structure although no special treatments are adopted in the problem formulation. This may be attributed to the ersatz material model and the stiffness interpolation scheme (i.e., (9)) used in the present work.

5.5 Extension to three dimension (3D) problems

The presented code can also be extended to solve 3D topology optimization problems. Let us consider a 3D version of the short beam example examined in Subsection 5.1. For this example, the length, width and the height of the design domain is 10, 5 and 1, respectively. For 3D problem, there are totally 9 geometry parameters (i.e., $\mathbf{D}^i=(x_{0i},y_{0i},z_{0i},l_i,w_i,t_i,\sin\alpha_i,\sin\beta_i,\sin\gamma_i)^T$) describing the geometry of a component with constant cross sectional area. The initial design, which

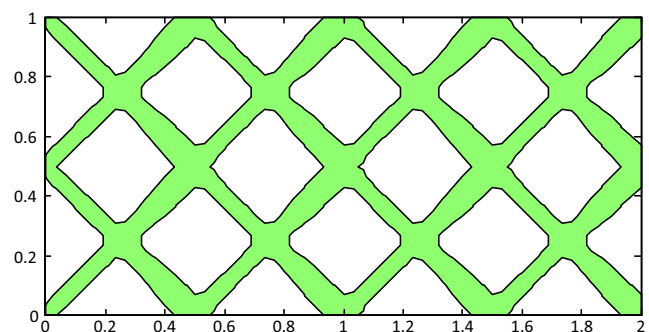


Fig. 12 The initial design for the compliant mechanism example

including 18 components, is shown in Fig. 14. The optimal topology obtained from a $40 \times 20 \times 4$ FEM mesh is shown in Fig. 15. The number of design variables under the MMC framework is $9 \times 18 = 162$ while in traditional framework is $40 \times 20 \times 4 = 3200$. Furthermore, if double resolution is required, the number of design variables will increase to 25,600 in traditional framework but it remains the same under the MMC framework. The readers can contact the authors directly to obtain the corresponding 3D code. More details

on the theoretical aspects of the 3D problem will be reported in a separate work.

6 Conclusions

In the present paper, a new topology optimization approach based on the so-called Moving Morphable Components (MMC) solution framework (Guo et al. 2014a) and the

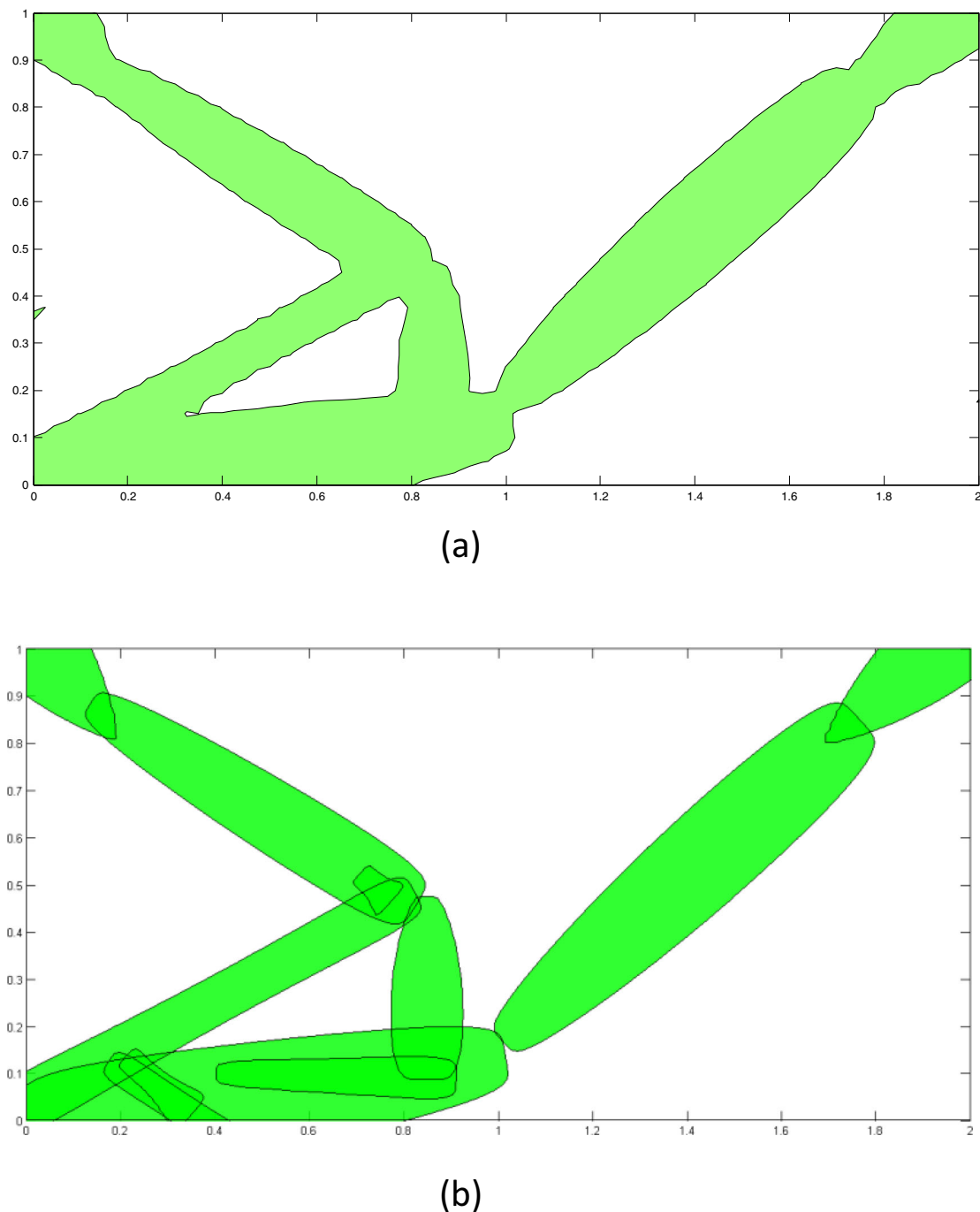


Fig. 13 Optimal topology of the compliant mechanism example: (a) the contour plot and (b) the component plot

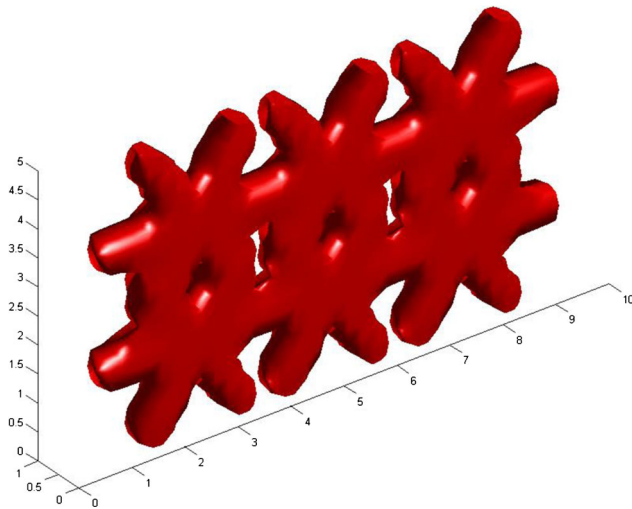


Fig. 14 The initial design for the 3D short beam example

ersatz material model is proposed. Compared with the existing MMC-based solution approach, the proposed method can not only allow for components with variable thicknesses but also enhance the numerical solution efficiency substantially. Numerical examples show the effectiveness and the robustness of the proposed approach. The proposed method can be implemented in a 188 line Matlab code and used to find optimized structural topologies with much less design variables. In order to help readers understand the essential features of the proposed approach, the Matlab code is also provided and disseminated in detail. It is also hoped that the publication of this code can encourage the readers to have an inside look of this MMC based approach and possibly allow them to make some refinements, extensions or modifications on this approach. Of course, the code is far from mature and many refined treatments of the MMC approach (Zhang et al. 2015a), both on the analysis and optimization aspects, are not reflected in it. The codes that implement several novel features of the MMC approach (e.g., X-FEM analysis based on accurate boundary

geometry description, sensitivity analysis based on accurate boundary/body integral, treatment of multi-phase material and three dimensional problems) will be released in our future work.

Acknowledgments The financial supports from the National Natural Science Foundation (10925209, 91216201, 11402048), the Fundamental Research Funds for the Central Universities, Program for Changjiang Scholars, Innovative Research Team in University (PCSIRT) and 111 Project (B14013) are gratefully acknowledged.

Appendix: Matlab code (corresponding to the short beam problem)

```
function MMC188(DW,DH,nelx,nely,x_int,
y_int,ini_val,volfrac)
% FEM data initialization
M=[ nely+1, nelx+1];
EW=DW / nelx; % length of element
EH=DH / nely; % width of element
[ x,y]=meshgrid(EW * [ 0 : nelx] , EH * [ 0 :
nely] );
LSgrid.x=x(:);
LSgrid.y=y(:); % coordinate of nodes
% Material properties
h=1; %thickness
E=1;
nu=0.3;
% Component geometry initialization
x0=x_int/2:x_int:DW; % x-coordinates of
the centers of components
y0=y_int/2:y_int:DH; % y-coordinates of
the centers of components
xn=length(x0); % number of component
groups in x direction
yn=length(y0); % number of component
groups in y direction
x0=kron(x0,ones(1,2*yn));
y0= repmat(kron(y0,ones(1,2)),1,xn);
N=length(x0); % total number of compo-
nents in the design domain
L= repmat(ini_val(1),1,N); % vector of the
half length of each component
t1= repmat(ini_val(2),1,N); % vector of
the half width of component at point A
t2= repmat(ini_val(3),1,N); % vector of
the half width of component at point B
t3= repmat(ini_val(4),1,N); % vector of
the half width of component at point C
st= repmat([ ini_val(5) -ini_val(5)] ,1,
N/2); % vector of the sine value of the in-
clined angle of each component
```

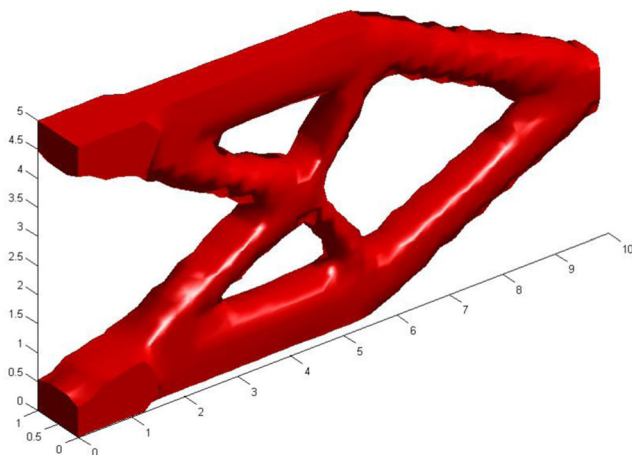


Fig. 15 Optimal topology of the 3D short beam example

```

variable=[ x0;y0;L;t1;t2;t3;st] ;
%Parameter of MMA
xy00=variable(:);
xval=xy00;
xold1=xy00;
xold2=xy00;
%Limits of variable:[ x0; y0; L; t1; t2; t3;
st] ;
xmin=[ 0; 0; 0.01; 0.01; 0.01; 0.03; -1.0] ;
xmin= repmat(xmin,N,1);
xmax=[ DW; DH; 2.0; 0.2; 0.2; 0.2; 1.0] ;
xmax= repmat(xmax,N,1);
low=xmin;
upp=xmax;
m=1; %number of constraint
Var_num=7; % number of design variables
for each component
nn=Var_num*N;
c=1000*ones(m,1);
d=zeros(m,1);
a0=1;
a=zeros(m,1);
%Define loads and supports (Short beam)
fixeddofs=1:2*(nely+1);
alldofs=1:2*(nely+1)*(nelx+1);
freedofs=setdiff(alldofs,fixeddofs);
loaddof=2*(nely+1)*nelx+nely+2;
F=sparse(loaddof,1,-1,2*(nely+1)*(nelx+1),1);
%Preparation FE analysis
nodenrs=reshape(1:(1+nelx)*(1+nely),1+nely,1+nelx);
edofVec=reshape(2*nodenrs(1:end-1,1:end-1)-1,nelx*nely,1);
edofMat=repmat(edofVec,1,8)+repmat([ 0 1 2*nelx+[ 2 3 4 5] 2 3],nelx*nely,1);
iK=kron(edofMat,ones(8,1))';
jK=kron(edofMat,ones(1,8))';
EleNodesID=edofMat(:,2:2:8)./2;
iEner=EleNodesID';
[ KE]=BasicKe(E,nu,EW,EH,h); % stiffness matrix k^s is formed
%Initialize iteration
p=6;
alpha=1e-3; % parameter alpha in the Heaviside function
epsilon=4*min(EW,EH); % regularization parameter epsilon in the Heaviside function
Phi=cell(N,1);
Loop=1;
change=1;
maxiter=1000; % the maximum number of iterations

while change>0.001 && Loop<maxiter
%Forming Phi^s
for i=1:N
Phi{ i}=tPhi(xy00(Var_num*i-Var_num+1:Var_num*i),LSgrid.x,LSgrid.y,p);
end
%Union of components
tempPhi_max=Phi{ 1} ;
for i=2:N
tempPhi_max=max(tempPhi_max,Phi{ i} );
end
Phi_max=reshape(tempPhi_max,nely+1,nelx+1);
%Plot components
contourf(reshape(x,M), reshape(y,M),Phi_max,[ 0,0] );
axis equal;axis([ 0 DW 0 DH] );pause(1e-6);
% Calculating the finite difference quotient of H
H=Heaviside(Phi_max,alpha,nelx,nely,epsilon);
diffH=cell(N,1);
for j=1:N
for ii=1:Var_num
xy001=xy00;
xy001(ii+(j-1)*Var_num)=xy00(ii+(j-1)*Var_num)+max(2*min(EW,EH),0.005);
tmpPhiD1=tPhi(xy001(Var_num*j-Var_num+1:Var_num*j),LSgrid.x,LSgrid.y,p);
tempPhi_max1=tmpPhiD1;
for ik=1:j-1
tempPhi_max1=max(tempPhi_max1,Phi{ ik} );
end
for ik=j+1:N
tempPhi_max1=max(tempPhi_max1,Phi{ ik} );
end
xy002=xy00;
xy002(ii+(j-1)*Var_num)=xy00(ii+(j-1)*Var_num)-max(2*min(EW,EH),0.005);
tmpPhiD2=tPhi(xy002(Var_num*j-Var_num+1:Var_num*j),LSgrid.x,LSgrid.y,p);
tempPhi_max2=tmpPhiD2;
for ik=1:j-1
tempPhi_max2=max(tempPhi_max2,Phi{ ik} );
end
for ik=j+1:N
tempPhi_max2=max(tempPhi_max2,Phi{ ik} );
end
end

```

```

HD1=Heaviside(tempPhi_max1,alpha,nelx,
nely,epsilon);
HD2=Heaviside(tempPhi_max2,alpha,nelx,
nely,epsilon);
diffH{ j} (:,ii) = (HD1-HD2) / (2* (max (2*min
(EW,EH),0.005)));
end
end
%FEA
denk=sum(H(ElNodesID).^2,2)/4;
den=sum(H(ElNodesID),2)/4;
A1=sum(den)*EW*EH;
U=zeros(2*(nely+1)*(nelx+1),1);
sK=KE(:)*denk(:)';
K=sparse(iK(:),jK(:),sK(:)); K=(K+
K')/2;
U(freedofs,:)=K(freedofs,freedofs)\F
(freedofs,:);
%Energy of element
energy=sum((U(edofMat)*KE).*U
(edofMat),2);
sEner=ones(4,1)*energy'/4;
energy_nod=sparse(iEner(:,1),sEner(:));
Comp=F'*U;
% Sensitivities
df0dx=zeros(Var_num*N,1);
dfdx=zeros(Var_num*N,1);
for k=1:N
df0dx(Var_num*k-Var_num+1:Var_num*k,
1)=2*energy_nod'.*H*diffH{ k};
dfdx(Var_num*k-Var_num+1:Var_num*k,1)
=sum(diffH{ k})/4;
end
%MMA optimization
f0val=Comp;
df0dx=-df0dx/max(abs(df0dx));
fval=A1/(DW*DH)-volfrac;
dfdx=dfdx/max(abs(dfdx));
[xmma,ymma,zmma,lam,xsi,eta,mu,zet,ss,
low,upp]=...
mmasub(m,nn,Loop,xval,xmin,xmax,xold1,
xold2,...
f0val,df0dx,fval,dfdx,low,upp,a0,a,c,
d);
xold2=xold1;
xold1=xval;
change=max(abs(xval-xmma));
xval=xmma;
xy00=round(xval*1e4)/1e4;
disp([' It.: ' sprintf('%4i\t',Loop) '
Obj.: ' sprintf('%6.3f\t',f0val) ' Vol.: ' ...
sprintf('%6.4f\t',fval) 'ch.: ' sprintf
('%6.4f\t',change)]);

```

```

Loop=Loop+1;
end
end
%Forming Phi_i for each component
function [ tmpPhi] =tPhi(xy,LSgridx,
LSgridy,p)
st=xy(7);
ct=sqrt(abs(1-st*st));
x1=ct*(LSgridx - xy(1))+st*(LSgridy - xy
(2));
y1=-st*(LSgridx - xy(1))+ct*(LSgridy -
xy(2));
bb=(xy(5)+xy(4)-2*xy(6))/2/xy(3)
^2*x1.^2+(xy(5)-xy(4))/2*x1/xy(3)+xy
(6);
tmpPhi=-((x1).^p/xy(3)^p+(y1).^p./bb.
^p-1);
end
%Heaviside function
function H=Heaviside(phi,alpha,nelx,
nely,epsilon)
num_all=[1:(nelx+1)*(nely+1)];
num1=find(phi>epsilon);
H(num1)=1;
num2=find(phi<=-epsilon);
H(num2)=alpha;
num3=setdiff(num_all,[num1;num2]);
H(num3)=3*(1-alpha)/4*(phi(num3)/
epsilon-phi(num3).^3/(3*(epsilon)^3))+
(1+alpha)/2;
end
%Element stiffness matrix
function[ KE]=BasicKe(E,nu,a,b,h)
k=[-1/6/a/b*(nu*a^2-2*b^2-a^2),1/8*nu
+1/8,-1/12/a/b*(nu*a^2+4*b^2-a^2),
3/8*nu-1/8,...
1/12/a/b*(nu*a^2-2*b^2-a^2),-1/8*nu-1/
8,1/6/a/b*(nu*a^2+b^2-a^2),-3/8*nu+1/
8];
KE=E*h/(1-nu^2)*...
[k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];
end
%~~~~ A Moving Morphable Components
(MMC) based topology optimization code
%~~~~ by Xu Guo Weisheng Zhang and Jie
Yuan

```

%~~~~ Department of Engineering
Mechanics, State Key Laboratory of
Structural Analysis
%~~~~ for Industrial Equipment, Dalian
University of Technology
%~~~~ Please send your suggestions and
comments to guoxu@dlut.edu.cn

References

- Allaire G, Olivier P (2006) Structural optimization with FreeFEM++. *Struct Multidiscip Optim* 32:173–181
- Allaire G, Jouve F, Toader AM (2004) Structural optimization using sensitivity analysis and a level-set method. *J Comput Phys* 194:363–393
- Andreassen E, Clausen A, Schevenels M, Lazarov B, Sigmund O (2011) Efficient topology optimization in matlab using 88 lines of code. *Struct Multidiscip Optim* 43:1–16
- Bendsoe MP (1989) Optimal shape design as a material distribution problem. *Struct Optim* 1:193–202
- Bendsoe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. *Comput Methods Appl Mech Engrg* 71:197–224
- Challis V (2010) A discrete level-set topology optimization code written in Matlab. *Struct Multidiscip Optim* 41:453–464
- Chen SK, Wang MY, Liu AQ (2008) Shape feature control in structural topology optimization. *Comput Aided Design* 40:951–962
- Eschenauer HA, Olhoff N (2001) Topology optimization of continuum structures: A review. *Appl Mech Rev* 54:331–390
- Guest J (2009a) Imposing maximum length scale in topology optimization. *Struct Multidiscip Optim* 37:463–473
- Guest J (2009b) Topology optimization with multiple phase projection. *Comput Method Appl Mech Eng* 199:123–135
- Guest J (2015) Optimizing the layout of discrete objects in structures and materials: a projection-based topology optimization approach. *Comput Methods Appl Mech Eng* 283:330–351
- Guest J, Prevost J, Belytschko T (2004) Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *Int J Numer Methods Eng* 61:238–254
- Guest J, Asadpoure A, Ha SH (2011) Eliminating beta-continuation from heaviside projection and density filter algorithms. *Struct Multidiscip Optim* 44:443–453
- Guo X, Cheng GD (2010) Recent development in structural design and optimization. *Acta Mech Sinica* 26:807–823
- Guo X, Zhao K, Wang MY (2005) A new approach for simultaneous shape and topology optimization based on dynamic implicit surface function. *Control Cybern* 34:255–282
- Guo X, Zhang WS, Zhong WL (2014a) Doing topology optimization explicitly and geometrically—a new moving morphable components based framework. *J Appl Mech* 81:081009
- Guo X, Zhang WS, Zhong WL (2014b) Explicit feature control in structural topology optimization via level set method. *Comput Method Appl Mech Eng* 272:354–378
- Guo X, Zhang WS, Zhang J (2015) Explicit structural topology optimization based on morphable components with complex shapes. In submission
- Ha SH, Guest JK (2014) Optimizing inclusion shapes and patterns in periodic materials using discrete object projection. *Struct Multidiscip Optim* 50:65–80
- Liu K, Tovar A (2014) An efficient 3D topology optimization code written in Matlab. *Struct Multidiscip Optim* 50:1175–1196
- Luo JZ, Luo Z, Chen SK, Tong LY, Wang MY (2008) A new level set method for systematic design of hinge-free compliant mechanisms. *Comput Method Appl Mech Eng* 198:318–331
- Michailidis G (2014) Manufacturing Constraints and multi-phase shape and topology optimization via a level-set method. Doctoral thesis Ecole Polytechnique <http://www.cmapx.polytechnique.fr/~michailidis/publis/thesis.pdf>
- Norato JA, Bellb BK, Tortorellic DA (2015) A geometry projection method for continuum-based topology optimization with discrete elements. *Comput Methods Appl Mech Eng* 293:306–327
- Otomori M, Yamada T, Izui K, Nishiwaki S (2015) Matlab code for a level set-based topology optimization method using a reaction diffusion equation. *Struct Multidiscip Optim* 51:1159–1172
- Petersson J, Sigmund O (1998) Slope constrained topology optimization. *Int J Numer Methods Eng* 41:1417–1434
- Poulsen TA (2003) A new scheme for imposing minimum length scale in topology optimization. *Int J Numer Methods Eng* 57:741–760
- Sigmund O (2001) A 99 line topology optimization code written in MATLAB. *Struct Multidiscip Optim* 21:120–127
- Sigmund O (2009) Manufacturing tolerant topology optimization. *Acta Mech Sinica* 25:227–239
- Sigmund O, Maute K (2013) Topology optimization approaches. *Struct Multidiscip Optim* 48:1031–1055
- Suresh K (2010) A 199-line Matlab code for Pareto-optimal tracing in topology optimization 42: 665–679
- Svanberg K (1987) The method of moving asymptotes—a new method for structural optimization. *Int J Numer Methods Eng* 24:359–373
- Wang MY, Wang XM, Guo DM (2003) A level set method for structural topology optimization. *Comput Methods Appl Mech Eng* 192:227–246
- Wang FW, Lazarov B, Sigmund O (2011) On projection methods, convergence and robust formulations in topology optimization. *Struct Multidiscip Optim* 43:767–784
- Xia Q, Shi TL (2015) Constraints of distance from boundary to skeleton: For the control of length scale in level set based structural topology optimization. *Comput Methods Appl Mech Eng*. doi:10.1016/j.cma.2015.07.015
- Xie YM, Steven GP (1993) A simple evolutionary procedure for structural optimization. *Comput Struct* 49:885–896
- Zhang WS, Zhong WL, Guo X (2014) An explicit length scale control approach in SIMP-based topology optimization. *Comput Methods Appl Mech Eng* 282:71–86
- Zhang WS, Zhang J, Guo X (2015a) Explicit structural topology optimization via moving morphable components- A revival of shape optimization. In submission
- Zhang WS, Zhong WL, Guo X (2015b) Explicit layout control in optimal design of structural systems with multiple embedding components. *Comput Methods Appl Mech Eng* 290:290–313
- Zhou M, Rozvany GIN (1991) The COC algorithm, part II: topological, geometry and generalized shape optimization. *Comput Methods Appl Mech Eng* 89:309–336
- Zhou MD, Lazarov BS, Wang FW, Sigmund O (2015) Minimum length scale in topology optimization by geometric constraint. *Comput Methods Appl Mech Eng* 293:266–28