A level set method for topological shape optimization of 3D structures with extrusion constraints

Hao Li [a], Peigen Li [a], Liang Gao [a], Li Zhang [a,*], Tao Wu [b]

[a] *State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China*
[b] *School of Software Engineering, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China*

Highlights

- We propose a new level set method for the topological shape optimization.
- Computational efficiency is much higher with the new parametric approach.
- The extrusion constraint is handled by a cross section projection strategy.
- Both single loading and multiple loading cases are well solved.

Abstract

This paper proposes a new level set method for topological shape optimization of 3D structures considering manufacturing constraints. First, the boundary of structure is implicitly represented as the zero level set of a higher-dimensional level set function, and the implicit surface is parameterized through the interpolation of a given set of compactly supported radial basis functions. In this way, the original Hamilton–Jacobi partial differential equation is transformed into a system of algebraic equations. Correspondingly, the topological shape optimization is changed to the easiest size optimization in structural optimization. Many more efficient gradient-based optimization algorithms can be directly applied to the size optimization. Second, to save the expensive computational cost in the 3D large-scale optimization problems, the discrete wavelet transform is introduced into the level set method to compress the size of the coefficient matrix in compactly supported radial basis function interpolant. The discrete wavelet transform converts the original matrix into a set of wavelet basis and therefore eliminates the "noise" elements in the new matrix, so that the linear system can be replaced by a sparser one. Finally, a cross section projection strategy is utilized to ensure the satisfaction of the extrusion constraint and reduce the number of the design variables, simultaneously. Several numerical examples in 3D structures are employed to demonstrate the effectiveness of the proposed method.

* Corresponding author. Tel.: +86 27 87557742; fax: +86 27 87543074.
 E-mail address: zlmse@mail.hust.edu.cn (L. Zhang).

1. Introduction

Topology optimization has been identified as one of the most promising as well as the most challenging disciplines in the field of structural optimization. In the numerical implementation, topology optimization automatically distributes a given amount of material in the fixed reference domain via an iterative manner, so as to optimize the objective function under specific constraints until the best layout of the structure is obtained. In recent years, structural topology optimization has gained remarkable development with a wide range of successful applications [1]. A number of methods have been established for solving these optimization problems, such as the homogenization method [2], the solid isotropic material with penalization (SIMP) approach [3,4], the point wise density-based interpolation (PDI) methods [5,6], the level-set based method [7–9], and the evolutionary structural optimization (ESO) method [10]. Topology optimization has been regarded as a powerful computational tool for the design of structures, mechanisms and materials. However, in the conceptual design of structures using topology optimization, another critical factor, manufacturability, should also be reasonably considered. Otherwise, it may result in an optimized design that cannot be fabricated.

Amongst many manufacturing features, the extrusion is a low-cost but an important aspect to produce structures with constant cross sections. It often minimizes the cost of secondary machining. This technique is capable of generating compressive and shear force in the product, and thus is widely used to make particular structures, such as dumpers of trucks, brackets mounting automobile engines, aircraft structural parts, window frame members, railings and so on. Generally, in the extrusion manufacturing process, materials are squeezed through an orifice of the fixed shape in a die by using the pressure from a ram, which is illustrated in Fig. 1. Hence, the key consideration in the design of an extrudable structure is to keep the identical cross sections along the extrusion path.

Due to the asymmetric or inconsonant boundary conditions, we cannot simplify a 3D structure with extrusion constraints to a 2D cross-sectional structure. Thus, some alternative methods have been developed for integrating the extrusion constraint into the topological design problem. For instance, Kim and Kim [11] studied the topology optimization problems considering the beam cross-section. Zhou et al. [12] proposed the mathematical formulation for topology optimization problems with extrusion constraint, which was implemented in the software Optistruct. Ishii and Aomura [13] utilized the homogenization method to solve the structural optimization problem considering the extrusion. Liu et al. [14] studied beam cross-section optimization problems considering warping of sections and coupling among deformations using SIMP approach. Patel et al. [15] proposed a method using the hybrid cellular automaton method to handle the extrusion-based nonlinear design problems. Zuberi et al. [16] investigated the influence of different configurations and locations of the load and boundary conditions for the extrudable designs. Yamada et al. [17] investigated the uniform cross-section surface constraints for the topology optimization problems, and proposed an efficient strategy by tuning the value of the regularization parameter. However, it can be found that most of the above studies were based on topology optimization methods of material distribution, which usually create optimized structures that are characterized with zigzag interfaces surrounded by intermediate material densities. To facilitate the fabrication of the design, additional post-processing schemes have to be employed to threshold element densities to enable a distinct, as well as a smooth interface.

The level set method [18–20] has emerged as a new alternative approach for topology optimization of structures since the work of Sethian and Wiegmann [7]. Recently, several different level set methods have been developed for topological shape optimization of structures, which can be categorized into two types. The first one is the conventional level set method with the common characteristics [7–9,21–23], such as the direct solution of the Hamilton–Jacobi partial differential equation (H–J PDE) using finite difference methods, a topological shape optimization process involving shape variations and topology changes simultaneously via the merging and splitting of the design boundary, a smooth geometric boundary and distinct material interface of the structure. Moreover, a physically meaningful solution of the H–J PDE can be ensured by the viscosity solution theory. However, in the conventional level set methods, the direct solution of the H–J PDE using the up-wind scheme is not an easy task, because several numerical aspects [7–9, 24–26] have to be carefully handled, including the CFL condition, re-initializations and velocity extension.

In response to the above numerical difficulties in the conventional level set methods, the other type of level set methods has been developed [27–34], without directly solving the H–J PDE using explicit finite difference methods. Particularly, the parametric level set method [24,29] may be one of the most effective and efficient level set methods for topological shape optimization of structures. The key concept of the parametric method is to discretize the implicit level set surface through an interpolation of the compactly supported radial basis functions (CSRBF). Since the CSRBFs are positioned at a given set of knots that are fixed in the design space, the H–J PDE is decoupled into
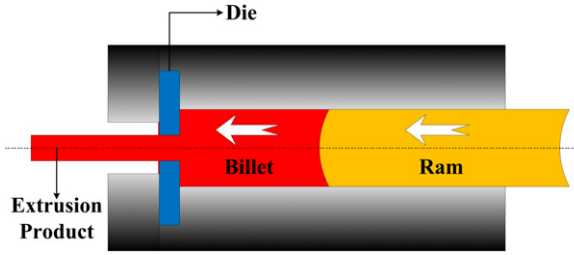
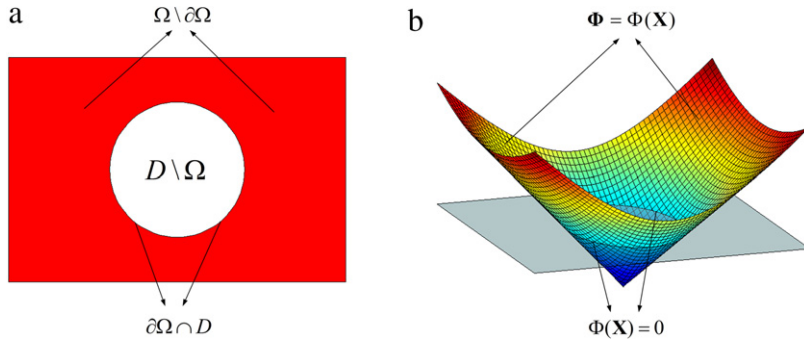Fig. 1. Description of the extrusion process in manufacturing.

Fig. 2. The level set-based structural boundary representation: (a) structural boundary; (b) level set surface.

more convenient ordinary differential equations (ODEs) and even a system of algebraic equations, in which only the coefficients of the interpolant are time-dependent and to be updated.

So far, there are only a limited amount of public research works about the topology optimization using level set methods for design problems considering manufacturing constraint [17,25]. Since the level set method is based on the concept of implicit dynamic surfaces, it is in nature more suitable for the representation of geometric features of manufacturing constraints for 3D structures. As a result, this paper aims to propose a new topological shape optimization method for design of 3D continuum structures subject to extrusion constraints, using a powerful parametric level set method with discrete wavelet transform (DWT) approximation.

2. A parametric level set method for topological shape optimization

2.1. Level set-based optimization model

In the level set-based structural optimization methods, the boundary of a structure is implicitly represented by a higher-dimensional level set function as its zero level set [18–20]. For instance, the boundary of a 2D structure is described by embedding it into the zero level set of a 3D scalar function that is defined on fixed Eulerian rectilinear grids (Fig. 2). Supposed that all the admissible shapes Ω are varying within a given reference domain D, the Lipschitz continuous level set function, as well as its zero level set, can be defined as follows:

$$\begin{cases} \Phi(\mathbf{x}) > 0, & \forall \mathbf{x} \in \Omega \setminus \partial\Omega \quad \text{(material)} \\ \Phi(\mathbf{x}) = 0, & \forall \mathbf{x} \in \partial\Omega \cap D \quad \text{(boundary)} \\ \Phi(\mathbf{x}) < 0, & \forall \mathbf{x} \in D \setminus \Omega \quad \text{(void)}. \end{cases} \tag{1}$$

The design of structure is governed by the dynamic front propagation in time, which simultaneously reflects shape variations of the geometric boundary, as well as topological changes of the structure via merging and splitting the design boundary. The motion of the boundary is mathematically modeled as a first-order H–J PDEs [8,9]. Introducing the pseudo-time *t* for deforming the structural boundary in time, and the corresponding H–J PDE is formulated as [18–20]:

$$\frac{\partial \Phi(\mathbf{x}, t)}{\partial t} - \mathbf{V_n} |\nabla \Phi| = 0. \tag{2}$$

In Eq. (2), it is noted that only the normal velocity field contributes the dynamic process of the design boundary, which is defined as follows:

$$\mathbf{V_n} = \mathbf{V} \cdot \mathbf{n} = \mathbf{V} \cdot \left(\frac{\nabla \Phi}{|\nabla \Phi|} \right) = \frac{d\mathbf{x}}{dt} \left(\frac{\nabla \Phi}{\sqrt{(\nabla \Phi)^2}} \right). \tag{3}$$

Therefore, the structural optimization can be regarded as a process of evolving the level set surface, as well as a process of updating the geometric boundary due to the normal velocity $\mathbf{V_n}$. It is noticed that the normal velocity field in the conventional level set methods is usually determined by the shape derivative analysis [35,36], which will be further detailed in the subsequent discussion.

In conventional level set methods, the level set function Φ is first discretized through a distance projection on a rectilinear Eulerian grid, as its general analytical form is usually unknown. Therefore, appropriate numerical method, such as the upwind scheme, has to be applied to solve the H–J PDE numerically [8,9]. However, as aforementioned, the direct solution of the H–J PDE is seldom easy. It is because that the numerical issues, such as the CFL condition, re-initializations to keep a signed distance function, and the extension of normal velocity field from the boundary to a narrow band or the entire design domain, have to be carefully handled in the optimization. These numerical features become unfavorable in the area of topological shape optimization, which will limit the application of the level set method to more advanced structural optimization problems [24–26].

2.2. CSRBF-based parametric level set method

The RBFs with global supports have been widely used as an effective tool to interpolate scattered data [37]. By positioning a family of radially-symmetric functions at the centers of a given set of N knots, the approximation of a function $f(X) : R^d \rightarrow R(d \geq 1)$ can be presented as

$$\tilde{f}(X) = \sum_{i=1}^{N} \alpha(X_i)\phi\left(\|X - X_i\|\right) \tag{4}$$

where X_i is the ith interpolation knot, and $\alpha(X_i)$ is the expansion coefficient. ϕ is the selected radially symmetric kernel. $\| \bullet \|$ denotes the Euclidean norm in a d-dimensional space.

In fact, despite the attractive features of the RBF interpolation, this method will always generate a linear system with a dense, ill-conditioned matrix, with the computational complexity of $O(N^k)$, $k = 2, 3$ [38]. This makes it quite difficult to apply RBFs to large-scale optimization problems. In addition, the shape parameter that has the great impact on the accuracy of the globally supported RBFs still lacks effective techniques to determine its optimal value [39]. Comparing with the globally supported RBFs, however, the CSRBFs [40,41] can lead to a sparse collection matrix. Furthermore, the CSRBF has the merit of positive definiteness, and the interpolation can naturally inherit the continuity of the CSRBF. In this paper, the Wendland C6 CSRBFs [40] is employed:

$$\phi(r) = (\max(0, 1 - r))^8 \cdot (32r^3 + 25r^2 + 8r + 1) \tag{5}$$

and according to the chain rule, the first-order derivatives of the CSRBFs are derived as follows:

$$\frac{\partial \phi}{\partial x} = \frac{\partial \phi}{\partial r} \cdot \frac{\partial r}{\partial x} = (\max(0, 1 - r)^7) \cdot (-352r^3 - 154r^2 - 22r) \cdot \frac{\partial r}{\partial x}, \tag{6}$$

$$\frac{\partial \phi}{\partial y} = \frac{\partial \phi}{\partial r} \cdot \frac{\partial r}{\partial y} = (\max(0, 1 - r)^7) \cdot (-352r^3 - 154r^2 - 22r) \cdot \frac{\partial r}{\partial y} \tag{7}$$

where the radius of support r defined in the two-dimensional Euclidean space is given as:

$$r = \frac{d_I}{R_I} = \frac{\sqrt{(x - x_i)^2 + (y - y_i)^2}}{R_I} \tag{8}$$

where d_I is the distance between any knots (x, y) inside the support and the current knot (x_i, y_i). R_I is used to control the size of the domain of influence of the basis function at the Ith knot. A proper radius of support can simultaneously ensure non-singularity of the interpolation and computational efficiency [29,40].

By introducing a series of CSRBFs and their expansion coefficients in the interpolation, the time-dependent dynamic level set function can be represented via an interpolation as follows:

$$\Phi(\mathbf{x}, t) = \boldsymbol{\varphi}(\mathbf{x})\boldsymbol{\alpha}^T(t) = \sum_{i=1}^{N} \phi_i(\mathbf{x})\alpha_i^T(t). \tag{9}$$

The vector of the CSRBFs is given by:

$$\boldsymbol{\varphi}(\mathbf{x}) = [\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \dots, \phi_N(\mathbf{x})] \in \mathbf{R}^N \tag{10}$$

and the expansion coefficients, which will act as the design variables, are given as:

$$\boldsymbol{\alpha}(t) = [\alpha_1(t), \alpha_2(t), \dots, \alpha_N(t)] \in \mathbf{R}^N. \tag{11}$$

We can simply express the linear system in Eqs. (9)–(11) with the matrix form:

$$\mathbf{A}\boldsymbol{\alpha}^T = \boldsymbol{\Phi} \tag{12}$$

where $\boldsymbol{\Phi} = [\Phi_1, \Phi_2, \dots, \Phi_N]^T$ represents the level set values over the N interpolation knots. $\boldsymbol{\alpha}$ is the vector consisting of the expansion coefficients. $\mathbf{A}$ is a theoretically invertible matrix with the size of $N \times N$, which can be given in terms of the CSRBFs:

$$\mathbf{A} = \begin{bmatrix} \boldsymbol{\varphi}(x_1) \\ \boldsymbol{\varphi}(x_2) \\ \dots \\ \boldsymbol{\varphi}(x_N) \end{bmatrix} = \begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) & \dots & \phi_N(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & \dots & \phi_N(x_2) \\ \dots & \dots & \dots & \dots \\ \phi_1(x_N) & \phi_2(x_N) & \dots & \phi_N(x_N) \end{bmatrix} \tag{13}$$

In the CSRBF-based level set method for topological shape optimization, we first need to solve the linear algebraic system in Eq. (12) to obtain the coefficient vector. This is because the initial level set values at CSRBF knots are actually pre-known, and the matrix $\mathbf{A}$ is computed once for all at the start of optimization loops. Then, in the subsequent iterations, we only focus on the product of matrix $\mathbf{A}$ and the vector $\boldsymbol{\alpha}$ in order to update the values of the level set function.

Thus, we can find that the CSRBFs are space-dependent only, and the expansion coefficients are time-dependent. Assuming that the CSRBF knots are located at the identical coordinates with the nodes of FEA mesh only for the simplicity, the approximation of the level set function is uniquely determined by the CSRBF associated with the expansion coefficients. Substituting the parameterized form of the level set function into Eq. (2), will yield the following equation:

$$\boldsymbol{\varphi}(\mathbf{x})\frac{d\boldsymbol{\alpha}^T(t)}{dt} - \mathbf{V_n}|\nabla\boldsymbol{\varphi}(\mathbf{x})\boldsymbol{\alpha}^T(t)| = 0 \tag{14}$$

and $\mathbf{V_n}$ is given by

$$\mathbf{V_n} = \frac{\boldsymbol{\varphi}(\mathbf{x})}{\left|\nabla\boldsymbol{\varphi}(\mathbf{x})\boldsymbol{\alpha}^T(t)\right|} \cdot \frac{d\boldsymbol{\alpha}^T(t)}{dt}. \tag{15}$$

This velocity field has been naturally extended to the whole design domain, where

$$\left|(\nabla\boldsymbol{\varphi}(x))\boldsymbol{\alpha}^T(t)\right| = \sqrt{\left(\frac{\partial\boldsymbol{\varphi}(\mathbf{x})}{\partial x}\boldsymbol{\alpha}^T(t)\right)^2 + \left(\frac{\partial\boldsymbol{\varphi}(\mathbf{x})}{\partial y}\boldsymbol{\alpha}^T(t)\right)^2}. \tag{16}$$

Hereto, the standard PDE-driven level set method has been transformed into a parametric form for solving the structural optimization problems. The parametric approach can simultaneously retain the beneficial characteristics of the standard level set method while avoid the unfavorable numerical issues, such as the direct solving of the H–J PDE using the up-wind scheme.

2.3. A matrix compression technique using DWT approximation

In this section, a DWT-based approximation method is developed to solve the system consisting of linear equations arising from the CSRBF interpolation. DWT, a multi-resolution decomposition for input data, has been widely used in signal processing, image compression, denoising and so on [42]. Recently, its application has been extended to handle large linear systems via compressing the relative matrices or vectors, due to its strong ability of rapidly capturing the significant information among a large amount of data with only a few coefficients [43,44]. Furthermore, a flexible thresholding process is always included, which will convert the dense coefficient matrix into a sparse pattern with expected accuracy, depending upon the discontinuities nature of data [45].

To transform an original basis into its corresponding wavelet basis, we first introduce the Haar wavelets with a constant scaling function and non-overlapping support [46,47], which are among the Daubechies families with compactly supported wavelets involving only one vanishing moments [46]. Then, by utilizing the fast wavelet transform algorithm [48], we can transform a vector or a matrix into a wavelet basis. It is noted that the requirement that the number of components in a vector must be a power of 2, when conducting the fast wavelet transform in traditional DWT application, has been extended to an arbitrary vector length [47].

To facilitate the illustration, we introduce a so-called "$\mathbf{W}$ matrix" [43] to implement the wavelet transform within the RBF interpolation. Given an orthogonal wavelet function in the continuous space, the relative orthogonal matrix $\mathbf{W}$ can transform the original basis to a new wavelet basis [44,45]. Considering the linear system in Eq. (12), and pre-multiplying the $\mathbf{W}$ matrix at its both sides gives:

$$\mathbf{W} \cdot \mathbf{A} \cdot \mathbf{W}^{\mathrm{T}} \cdot \mathbf{W} \cdot \boldsymbol{\alpha}^{\mathrm{T}} = \mathbf{W} \cdot \boldsymbol{\Phi}. \tag{17}$$

On the other hand, multiplying with the $\mathbf{W}$ matrix on the vector $\boldsymbol{\alpha}$ and $\boldsymbol{\Phi}$ yields:

$$\hat{\boldsymbol{\alpha}} = \mathbf{W} \cdot \boldsymbol{\alpha} \tag{18}$$

$$\hat{\boldsymbol{\Phi}} = \mathbf{W} \cdot \boldsymbol{\Phi}. \tag{19}$$

For the two dimensional transform of matrix $\mathbf{A}$, the wavelet basis can be obtained as:

$$\hat{\mathbf{A}} = \mathbf{W} \cdot \mathbf{A} \cdot \mathbf{W}^{\mathrm{T}}. \tag{20}$$

By substituting Eqs. (18)–(20) into Eq. (17), we can establish the new system as:

$$\hat{\mathbf{A}} \cdot \hat{\boldsymbol{\alpha}}^{\mathrm{T}} = \hat{\boldsymbol{\Phi}}. \tag{21}$$

After the transformation of the linear system, only a part of elements with large absolute values are useful in the new matrix $\hat{\mathbf{A}}$, due to the smoothness of original matrix $\mathbf{A}$ in terms of CSRBFs. Thus, the smaller coefficients in the matrix $\hat{\mathbf{A}}$ will not be considered by using the hard thresholding. This means we can obtain a highly compressed matrix $\hat{\mathbf{A}}^*$ from $\hat{\mathbf{A}}$ after thresholding. As a result, we actually have a much sparser system of the linear equations, compared to the original matrix $\mathbf{A}$:

$$\hat{\mathbf{A}}^* \cdot \hat{\boldsymbol{\alpha}}^{\mathrm{T}} = \hat{\boldsymbol{\Phi}}. \tag{22}$$

When calculating the vector of $\hat{\boldsymbol{\alpha}}$, a direct or iterative approach can be utilized [43]. On the other hand, to determine the vector of $\hat{\boldsymbol{\Phi}}$, one can multiply the sparse version of $\hat{\mathbf{A}}^*$ and $\hat{\boldsymbol{\alpha}}$ straightforwardly. Once $\hat{\boldsymbol{\alpha}}$ or $\hat{\boldsymbol{\Phi}}$ is obtained, we can convert this wavelet basis into its original form to achieve the final solution vector via the inverse transform:

$$\boldsymbol{\alpha} = \mathbf{W}^{\mathrm{T}} \cdot \hat{\boldsymbol{\alpha}} \quad \text{or} \quad \boldsymbol{\Phi} = \mathbf{W}^{\mathrm{T}} \cdot \hat{\boldsymbol{\Phi}}. \tag{23}$$

Hence, the main advantage of this DWT procedure is that it can increase the sparseness of large-scale linear systems to reduce the computational cost. That is to say, it can produce an accurate solution only with a small amount of elements in the matrix $\hat{\mathbf{A}}^*$. Furthermore, even if the new system in Eq. (22) is not accurate enough, Eq. (23) can still provide a good approximation of $\boldsymbol{\alpha}$ or $\boldsymbol{\Phi}$ [43,47].

3. A cross-section projection method

An extrudable design means a design with constant cross-sections along the extrusion direction, which has been seldom studied in the area of topology optimization with level sets. To ensure a final design, which can be fabricated

by the process of extrusion, a large number of constraints must be explicitly introduced into the optimization problem to ensure that all the nodes on the given axis have the same values of discrete level set. Thus, extra efforts and complicated solution techniques have to be elaborately applied to account for the manufacturing constraints [11,12]. Nevertheless, in the present study, we aim to simplify the optimization model subject to the extrusion constraint, and propose a cross-sectional projection method to map each individual element in the 3D space to a corresponding projected plane. It is expected that the original 3D design problem is converted to a problem of identifying the optimal material distribution within a 2D space with a small number of design variables.

For the sake of numerical simplicity but without losing any generality, in this paper, the widely-studied and well-recognized mean compliance problem in the area of topology optimization is employed to demonstrate the effectiveness of the proposed method. In general, the mean compliance optimization problem with the extrusion constraint can be formulated as:

$$
\begin{aligned}
&\text{Find: } \boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]\\[4pt]
&\underset{(u, \Phi)}{\text{Minimize}} : J(u, \Phi) = \int_D f(u) H(\Phi) d\Omega\\[4pt]
&\text{Subject to:} \quad V(\Phi) = \int_D H(\Phi) d\Omega - V_{max} \leq 0\\
&\qquad\qquad\quad a(u, \partial u, \Phi) = l(\partial u, \Phi), u|_{\partial\Omega} = u_0, \quad \forall \partial u \in U\\
&\qquad\qquad\quad \left(\Phi_i = \Phi_j = \cdots = \Phi_{NE}\right)_k, \quad k = 1, 2, \ldots, M
\end{aligned}
\tag{24}
$$

where J is the objective functional and V is the constraint of material usage with an upper bound of V_{max}, respectively. H is the Heaviside function associated with the level set function Φ. M denotes the number of elements in a single cross section, and NE represents the number of elements along the extrusion path. $\Phi_i = \Phi_j = \cdots = \Phi_{NE}$ are considered as the manufacturing constraints.

As indicated above, the cross-sectional projection method is actually a heuristic scheme, of which the main procedures can be summarized as follows:

Step 1: *Aggregate the individual elements in* 3D *meshes onto a specified* 2D *plane.*
Step 2: *Update the design variables and the level set functions in the* 2D *domain.*
Step 3: *Renew the design boundary in the original* 3D *model.*

It is easy to see that Step 2 can be achieved by the present parametric level set method in the reduced set, while Step 3 is accomplished through updating the discrete values of the level set function in the 2D domain along the extrusion path to map the original 3D grids. The difficulty now is how to appropriately handle the elements within the 3D model to achieve Step 1.

To solve the above issue, in this paper, we define two types of elements to achieve the mapping of the 3D model to 2D. Let us consider an arbitrary element E_i in the discretized 3D design space. Supposing that the extrusion path is along the **Z** axis, two types of elements in the FE model that influence E_i are named as the adjacent elements and the parallel elements, from the point of view of extrusion, which are shown in Fig. 3(a) and (b). The adjacent elements are defined as the elements surrounding E_i in a pre-determined sub-domain. These elements influence the connection of E_i to the transmission path of the load, and are mainly used to prevent mesh-dependence in the optimal design. On the other hand, the parallel elements, which are specified as elements in the same extrusion path of E_i, are utilized to ensure the extrudable design by maintaining the constant level-set values among them.

Mathematically, two specified formulas, i.e. Eqs. (25) and (26), can be used to account for the influences from the two types of elements to achieve Step 1. For instance, the bilinear functional associated with 2D domain can be expressed as follows:

$$
a_P(u, \partial u, \Phi) = \sum_{k=1}^{M} \left\{ \sum_{i=1}^{NE} \bar{a}(u, \partial u, \Phi)_i / NE \right\}_k
\tag{25}
$$

$$
\bar{a}(u, \partial u, \Phi)_i = \frac{1}{\sum\limits_{j=1}^{NAE} w_{ij}} \sum_{j=1}^{NAE} \left[w_{ij} \left(\int_{D_{Ei}} (\varepsilon(\partial u_i))^{\mathrm{T}} C(E_i) \varepsilon(u_i) H(\Phi_i) d\Omega_{Ei} \right) \right],
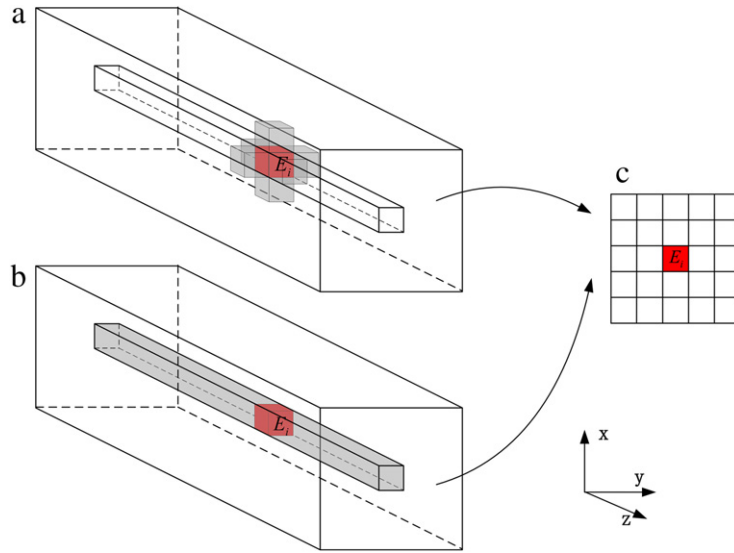\tag{26}
$$

$$
w_{ij} = rd - dist(i, j)
$$

Fig. 3. Schematic description of the cross section projection strategy: (a) the adjacent elements; (b) the parallel elements; (c) the projection plane.

where Eq. (25) is used to average the influences from the elements in the identical axis with E_i, and Eq. (26) is implemented to aggregate the influences from the neighbor elements of E_i. Ω_{Ei} is the domain occupied by individual element E_i. The bilinear functional $\bar{a}(u, \partial u, \Phi)_i$ in a sub-domain Ω_{Ei} is given in the weak form. rd is the pre-determined radius of neighborhood, and *dist* is the distance between E_i and its *j*th adjacent element. w_{ij} is the weight coefficient that is defined as the difference between rd and *dist*. *NAE* represents the number of adjacent elements within the fixed radius *rd*.

Similarly, the loading functional is derived as:

$$l_P(\partial u, \Phi) = \sum_{k=1}^{M} \left\{ \sum_{i=1}^{NE} \bar{l}(\partial u, \Phi)_i / NE \right\}_k \tag{27}$$

$$\bar{l}(\partial u, \Phi)_i = \frac{1}{\sum\limits_{j=1}^{NAE} w_{ij}} \sum_{j=1}^{NAE} \left[w_{ij} \left(\int_{D_{Ei}} p \partial u H(\Phi_i) d\Omega_{Ei} + \int_{\Gamma_{Ei}} \tau \partial u H d\Gamma_{Ei} \right) \right], \tag{28}$$

$$w_{ij} = rd - dist(i, j).$$

Then, after the projection, we obtain a new formulation that is described in a 2D space Ω_P:

$$\underset{(u, \Phi)}{\text{Minimize: }} J_P(u, \Phi) = \int_{D_P} f(u) H(\Phi) d\Omega_P$$

$$\text{Subject to: } \begin{cases} V_P(u, \Phi) = \int_{D_P} H(\Phi) d\Omega_P - V_P^{max} \leq 0 \\ a_P(u, \partial u, \Phi) = l_P(\partial u, \Phi), u|_{\partial \Omega_P} = u_0, \quad \forall \partial u \in U_P. \end{cases} \tag{29}$$

Comparing with Eq. (24), the manufacturing constraints are embedded implicitly into the formulation of Eq. (29). However, this formulation can ensure a topological shape optimization design to facilitate the extrusion fabrication. In this way, the number of design variables in Eq. (29) has been greatly reduced when the extrusion constraint is satisfied. Moreover, the number of the interpolation operators with regard to the parametric level set method is accordingly reduced.

4. Shape derivative analysis and optimization algorithm

Here, the shape sensitivity analysis [35,36] is applied to calculate the derivative of a behavior functional with regard to the expansion coefficients of the CSRBF interpolant.

For a constrained optimization problem formulated in Eq. (29), we introduce the Lagrangian multipliers λ_0 and λ_1, which are associated with the volume constraint and the state equation, to derive an unconstrained problem accordingly. The Lagrangian functional is given as:

$$L_P(u, \Phi) = J_P(u, \Phi) + \lambda_0 V_P(u, \Phi) + \lambda_1 [l_P(\partial u, \Phi) - a_P(u, \partial u, \Phi)]. \tag{30}$$

With the concept of material derivatives [35,36], the shape derivative of the Lagrangian functional is:

$$\frac{\partial L_P(u, \Phi)}{\partial t} = \int_{D_P} \beta(u, \partial u, \Phi) |\nabla \Phi| \mathbf{V_n} \delta(\Phi) d\Omega_P = \int_{\Gamma_P} \beta(u, \partial u, \Phi) \mathbf{V_n} d\Gamma_P \tag{31}$$

where the shape gradient density relevant to the 2D projected design domain is obtained via the following two steps, according to the particular operations described in Section 3:

$$\beta(u, \partial u, \Phi) = \sum_{i=1}^{NE} \bar{\beta}(u, \partial u, \Phi)_i / NE \tag{32}$$

$$\bar{\beta}(u, \partial u, \Phi)_i = \frac{1}{\sum_{j=1}^{NAE} w_{ij}} \sum_{j=1}^{NAE} \left\{ w_{ij} \left[f(u) - \varepsilon^{\mathrm{T}}(\partial u) E \varepsilon(u) + (p\partial u + \nabla(\tau \partial u) \cdot \mathbf{n} + \nabla \cdot \mathbf{n} \cdot \tau \partial u) \right] \right\} + \lambda_0, \tag{33}$$

$$w_{ij} = rd - dist(i, j)$$

where $\bar{\beta}$ denotes the elementary shape gradient density under the influences of the adjacent elements in the three-dimensional space. β is the corresponding shape gradient density in the 2D projected domain.

Recalling $\mathbf{V_n}$ given in Eq. (15) and substituting it into Eq. (31), the shape derivative of the Lagrangian functional can be rewritten by:

$$\frac{\partial L_P(u, \Phi)}{\partial t} = \sum_{k=1}^{M} \int_{\Gamma_P} \left(\beta(u, \partial u, \Phi) \frac{\phi_k(\mathbf{x})}{\left| \nabla \boldsymbol{\varphi}(\mathbf{x}) \boldsymbol{\alpha}^{\mathrm{T}}(\mathbf{t}) \right|} \right) \frac{d\boldsymbol{\alpha}(t)}{dt} d\Gamma_P \tag{34}$$

where Eq. (34) can be divided into two components as follows:

$$\frac{\partial L_P(u, \Phi)}{\partial t} = \sum_{k=1}^{M} \int_{\Gamma_P} \left(\vartheta(u, \partial u, \Phi) \frac{\phi_k(x)}{\left| \nabla \boldsymbol{\varphi}(\mathbf{x}) \boldsymbol{\alpha}^{\mathrm{T}}(t) \right|} \right) \frac{d\boldsymbol{\alpha}(\mathbf{t})}{dt} d\Gamma_P + \lambda_0 \sum_{k=1}^{M} \int_{\Gamma_P} \frac{\phi_k(x)}{\left| \nabla \boldsymbol{\varphi}(\mathbf{x}) \boldsymbol{\alpha}^{\mathrm{T}}(t) \right|} \frac{d\boldsymbol{\alpha}(t)}{dt} d\Gamma_P \tag{35}$$

where ϑ is the rest part of the shape gradient density β excluding the Lagrangian multiplier λ0.

On the other side, using the chain rule, the shape derivative of the Lagrangian functional is written as

$$\frac{\partial L_P(u, \Phi)}{\partial t} = \sum_{k=1}^{M} \frac{\partial J_P(u, \Phi)}{\partial \alpha_k(t)} \frac{d\boldsymbol{\alpha}(\mathbf{t})}{dt} + \lambda_0 \sum_{k=1}^{M} \frac{\partial V_P(u, \Phi)}{\partial \alpha_k(t)} \frac{d\boldsymbol{\alpha}(t)}{dt}. \tag{36}$$

Comparing Eqs. (35) and (36), the design sensitivities associated with the design variables are:

$$\frac{\partial J_P(u, \Phi)}{\partial \alpha_k(t)} = \int_{\Gamma_P} \vartheta(u, \partial u, \Phi) \frac{\phi_k(\mathbf{x})}{\left| \nabla \boldsymbol{\varphi}(\mathbf{x}) \boldsymbol{\alpha}^{\mathrm{T}}(t) \right|} d\Gamma_P \tag{37}$$

$$\frac{\partial V_P(u, \Phi)}{\partial \alpha_k(t)} = \int_{\Gamma_P} \frac{\phi_k(\mathbf{x})}{\left| \nabla \boldsymbol{\varphi}(\mathbf{x}) \boldsymbol{\alpha}^{\mathrm{T}}(t) \right|} d\Gamma_P. \tag{38}$$

Here, $k = 1, 2, \ldots, M$, where M is the number of design variables for the proposed optimization model. It is obvious that the number of M is far less than the number of CSRBF knots in three-dimensional design space. This enables us to control the moving boundary merely in the projection domain.
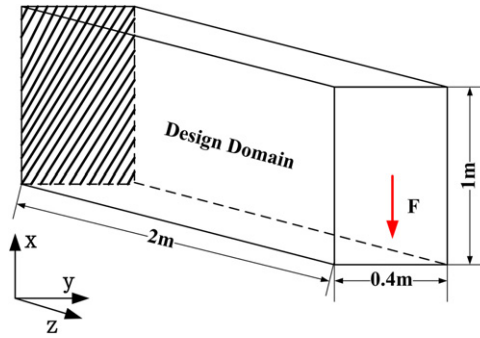
Fig. 4. Design domain for the single loading case.

In order to improve the numerical efficiency of the optimization, we transform the design sensitivity expressed by the boundary integration scheme into the following volume integration [24,29]:

$$\frac{\partial J_P(u, \Phi)}{\partial \alpha_k(t)} = \int_{D_P} \vartheta(u, \partial u, \Phi)\phi_k(x)\delta(\Phi)d\Omega_P \tag{39}$$

$$\frac{\partial V_P(u, \Phi)}{\partial \alpha_k(t)} = \int_{D_P} \phi_k(x)\delta(\Phi)d\Omega_P. \tag{40}$$

Here, the Delta function $\delta(\Phi)$ is stated as:

$$\delta(\Phi) = \frac{1}{\pi}\frac{\eta}{\Phi^2 + \eta^2} \tag{41}$$

where η is usually a positive constant selected as 2–4 times as the mesh size.

Once the design sensitivities of the objective functional and volume constraint are given in Eqs. (39) and (40), we can make use of many well-established optimization algorithms to update the design variables, e.g., the optimality criteria (OC) method [49] and the method of moving asymptotes (MMA) [50]. The OC method is more efficient for the optimization problems with a large number of design variables but single constraint. So the OC method will be used in this paper [29,49].

5. Numerical examples

Here, several typical numerical examples are used to demonstrate the characteristics of the proposed method. In these examples, the popularly used "ersatz material" model [9] is implemented to calculate the strain energy of boundary crossing elements without remeshing. The level set function is initially embedded as a signed distance function without further re-initializations. The CSRBF knots are assumed to be identical to the FEA nodes for the sake of numerical simplicity. The CSRBFs with C6 continuity [40] are applied to all the numerical cases. The Young's modulus of material is 180 GPa and Poisson's ratio is 0.3. In all the 3D cases, the design domains are discretized with the 8-node tri-linear cube finite elements. All the MATLAB routines are operated on a desktop computer with up-to-date standard configurations.

5.1. Optimization problem with single loading case

In this section, we investigate the optimization problem of a cantilever beam under single load. The design problem of the 3D structure is shown in Fig. 4. The entire design domain is a cuboid solid with a dimension of 1 m × 0.4 m × 2 m, and only half of the design domain (i.e. 1 m × 0.2 m × 2 m) is optimized considering the symmetry. The left end is fixed and a vertical unit external force of magnitude 300 kN is applied at the center of the free end. The objective is to minimize the total strain energy of the structure for a given amount of material usage (50%).

5.1.1. Investigation of the thresholding factor

Here, we examine the effect of thresholding factor in the DWT scheme. The entire design domain is discretized into $20 \times 4 \times 40 = 3200$ finite element meshes. Correspondingly, a set of uniformly distributed CSRBF knots ($21 \times 5 \times 41 =$

Table 1
Comparison of GSRBF, CSRBF and CSRBF+DWT with different κ values.

Case	κ	Size of matrix A	Number of zeros	Percentage of zeros	Fitting time (s)	Evaluation time/Step (s)
GSRBF	0	4305×4305	0	0%	6.221	0.041
CSRBF	0	4305×4305	16 715 800	90.19%	1.031	0.015
CSRBF+DWT (a)	0.1	4305×4305	17 356 510	93.65%	1.618	0.019
CSRBF+DWT (b)	1	4305×4305	17 702 904	95.52%	1.302	0.017
CSRBF+DWT (c)	10	4305×4305	18 175 024	98.07%	0.617	0.015
CSRBF+DWT (d)	20	4305×4305	18 303 790	98.76%	0.554	0.012
CSRBF+DWT (e)	100	4305×4305	18 496 282	99.80%	0.241	0.011

4305) is used to interpolate the level set function, which will lead to a large system of linear equations with the size of the interpolant matrix "$\mathbf{A}$" up to 4305×4305. As mentioned above, the key of the DWT is to produce a sparser matrix to replace the original interpolation matrix by denoising the elements with small absolute values in the transformed wavelet matrix. Here, the hard thresholding strategy given in [47] is used to zero out the redundant elements:

$$\hat{\phi}_{ij}^* = \begin{cases} \hat{\phi}_{ij}, & \left| \hat{\phi}_{ij} \right| \geq \gamma \\ 0, & \text{otherwise} \end{cases} \tag{42}$$

where the threshold is governed by $\gamma = \kappa \bar{n}$. $\bar{n}$ is the average of absolute values regarding the elements in $\hat{\mathbf{A}}$. $\hat{\phi}_{ij}$ are the coefficients in the wavelet basis $\hat{\mathbf{A}}$, and $\hat{\phi}_{ij}^*$ are the coefficients in the sparse matrix $\hat{\mathbf{A}}^*$, respectively. The factor $\kappa \geq 0$ is introduced to tune the magnitude of the threshold.

To show the numerical efficiency, we applied GSRBF, CSRBF and CSRBF+DWT with different κ values to the same linear system. The numerical processes for computing $\boldsymbol{\alpha}$ and $\boldsymbol{\Phi}$ in Eq. (12) are named as fitting and evaluation, respectively. The results in Table 1 show that when using the GSRBF (i.e. Gaussian kernel, Multiquadric kernel and Inverse multiquadric kernel) a fully dense matrix "$\mathbf{A}$" is obtained. For the CSRBF, approximately 90% zero elements are found in the matrix. Comparing with the GSRBF-based interpolation, this will significantly reduce the requirement of computer memory, as well as the computational time (i.e. the fitting time is reduced from 6.221 to 1.031 s, and the evaluation time is reduced from 0.041 to 0.015 s per step).

From the 3rd to the 7th row in Table 1, we incorporate the matrix compression technique DWT into CSRBF. Different values of factor κ are adopted. It suggests that a larger κ will result in a better matrix compressing ratio but loss of interpolation accuracy and vice versa. This means a proper factor κ can simultaneously satisfy the computational efficiency and the numerical accuracy. From the results we can find that the larger value of κ can reduce the computational cost due to the sparseness of the matrix $\mathbf{A}$. It is noted that when $\kappa = 0.1$ or 1, the computational time of CSRBF+DWT is greater than that of the direct implementation of CSRBF. This is because it takes some extra CPU time to construct the new system, as given in Eq. (22). It is obvious that the percentage of zeros is up to 99.80% and the CPU times for fitting and evaluation reduced dramatically when $\kappa = 100$. However, the numerical accuracy will be greatly decreased. Hence, it can be found that an approximate number of 99% zero elements can give us a good tradeoff between sparseness and accuracy in these linear systems, and the factor $\kappa = 20$ is adopted in this study. We further investigate the numbers of zeros with different candidate knots, whose results are shown in Table 2. It can be found that with a factor $\kappa = 20$, the CSRBF+DWT based approach can always reach a near 99% sparseness.

5.1.2. Investigation of the DWT scheme

In this section, we illustrate the role of the DWT scheme in the parametric level set formulation to solve the topological shape optimization problem. The same finite element mesh ($20 \times 4 \times 40$) is still used to discretize the design domain, as shown in Fig. 4. For comparison, we first parameterize the level set function with the CSRBF incorporating the DWT, and second only using the CSRBF, which are denoted as Case 1 and Case 2. The OC method has a small move limit (1×10^{-5}) in order to stabilize the iteration when an extremely sparse matrix $\mathbf{A}$ is involved. The damping factor is 0.5, and the terminal criterion is that the difference of two successive objective values is less than 1×10^{-3} or the maximum iteration (400) is reached.

Fig. 5 shows the identical initial design that is used for the two different cases and their corresponding optimal configurations. In each case, the favorable features of the level set methods are well remained, such as the smooth

Table 2
Comparison of numbers of zero elements in different knots pattern.

Method	κ	Knots pattern	Size of matrix A	Number of zeros	Percentage of zeros
GSRBF	0	$21 \times 5 \times 41$	4305×4305	0	0%
CSRBF	0	$21 \times 5 \times 41$	4305×4305	16 715 800	90.19%
CSRBF+DWT	20	$21 \times 5 \times 41$	4305×4305	18 303 790	98.76%
GSRBF	0	$21 \times 5 \times 81$	8505×8505	0	0%
CSRBF	0	$21 \times 5 \times 81$	8505×8505	68 632 240	94.88%
CSRBF+DWT	20	$21 \times 5 \times 81$	8505×8505	71 618 424	99.01%
GSRBF	0	$21 \times 9 \times 81$	$15 309 \times 15 309$	0	0%
CSRBF	0	$21 \times 9 \times 81$	$15 309 \times 15 309$	225 207 870	96.09%
CSRBF+DWT	20	$21 \times 9 \times 81$	$15 309 \times 15 309$	232 568 456	99.23%

Table 3
Numerical results for Case 1 and Case 2.

Case	Method	κ	Knots pattern	Number of steps	Time per step (s)	Optimal compliance (N mm)
1	CSRBF+DWT	20	$21 \times 5 \times 41$	241	160.712	176.580
2	CSRBF	0	$21 \times 5 \times 41$	400	175.553	179.437

boundary, distinct interfaces, integrated shape and topology optimizations [18,19]. The only slight difference of these two designs with respect to the topologies is the middle parts of the optimal structures. This is because that Case 2 is converged to a different local minimum due to the design not being fully evolved within the allowable 400 iterations.

The iteration histories for the two cases are shown in Fig. 6, and the numerical results are given in Table 3. For each case, we found that the iteration is quite stable. The objective functionals for the two cases increase at the beginning stage, due to the violation of the constraint. After that, the material constraint is gradually satisfied, and the structural compliance starts to decrease. However, it is obvious that Case 1 has a faster convergence than Case 2, which indicates a great saving of the computational cost. Besides, the optimal results of Case 1 are slightly stiffer than that of Case 2.

From these results, we can see that the DWT scheme can be combined with the parametric level set method, to reduce the computational cost, while improve the numerical efficiency at the same time. It is noted that the proposed method has an influence to the numerical accuracy during iterations, but the overall numerical accuracy of the optimization can be reasonably acceptable.

5.1.3. Investigation of the cross section projection strategy

To consider the extrusion constraint in the design problem given in Fig. 4, we now incorporate the cross section projection strategy into the framework of the parametric level set method. The FEA model and the level set grids remain unchanged as the previous examples. The extrusion path is enforced along the **Y** axis direction.

The initial design is quite different from that of the case without the extrusion constraint. It contains several holes crossing the entire structure along the extrusion direction to maintain the uniform cross section, which is shown in Fig. 7(a). The optimal configuration is plotted in Fig. 7(b). Comparing with the optimal designs given in Fig. 5, we can find that the cross sections in this case are constant. This is because the value of level set function at each knot with the same coordinate in **XZ**-plane is restrained to be identical by the operator upon parallel elements.

The convergence is shown in Fig. 8. It takes 136 steps to minimize the structural strain energy from 480.045 to 178.904. It can be found that the numerical process is stable, and the volume constraint is conservative. Comparing with the result of Case 1 in Table 3, the design without the extrusion constraint is somewhat stiffer than the result with the extrusion constraint, which may be because of the extrusion constraint narrowing down the search space.

Comparing with the cases with the full 3D simulations (e.g. Case 1 and Case 2 in Table 3), the average CPU time for the iterative process has been reduced noticeably to 113.712 s per step. The reason is that the 3D optimization problem with the extrusion constraint has been transformed into a size optimization issue within the much smaller 2D projection domain (21×41 knots). Hence the proposed method can ensure much less computational cost for an extrudable topology design.
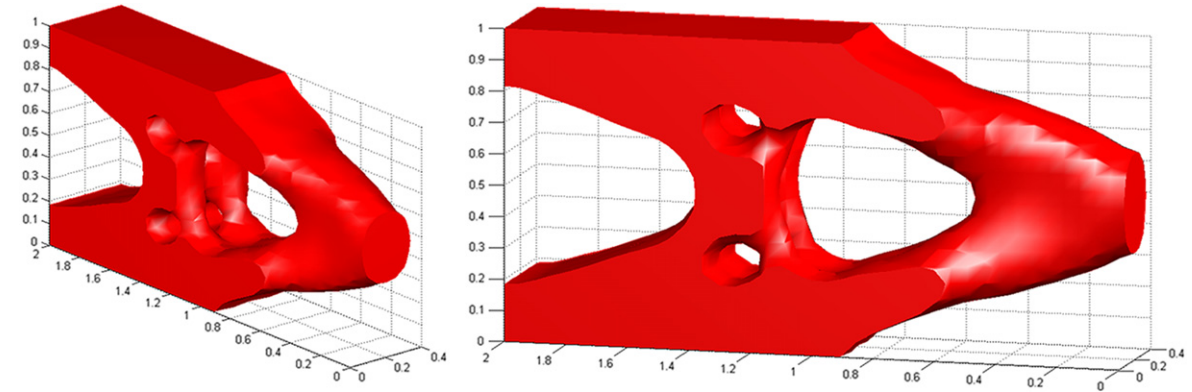
Furthermore, to show that the proposed method is free from mesh-dependence due to the consideration of the influences of the adjacent elements, three numerical tests with different mesh sizes for the same design problem shown
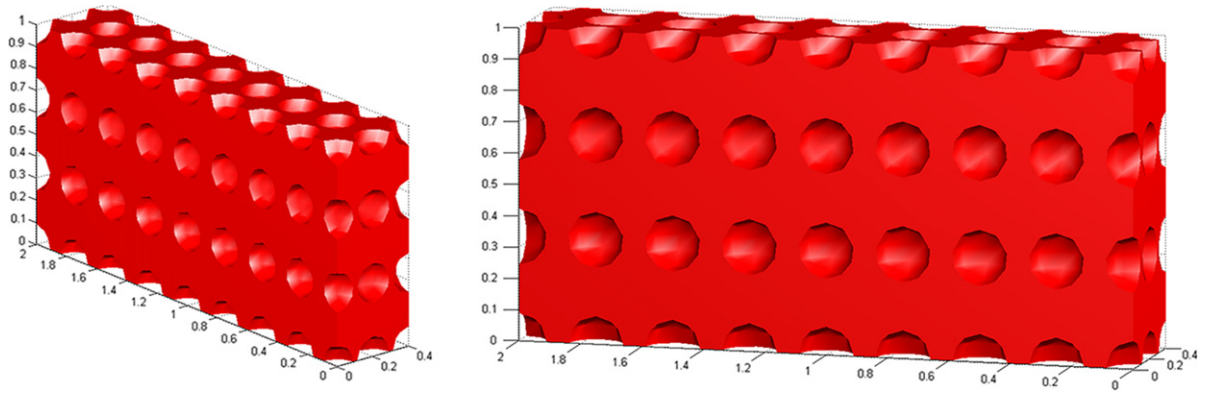
(a) Initial design.
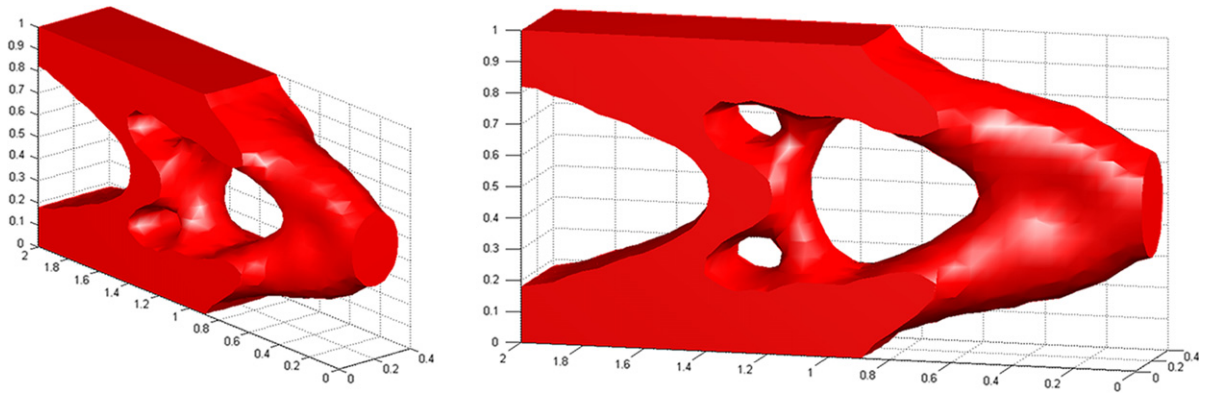
(b) Optimal design of Case 1.

(c) Optimal design of Case 2.

Fig. 5. Initial design and optimal topologies for the single loading case without extrusion constraint: (a) initial design; (b) Case 1: with CSRBF and DWT; (c) Case 2: only with CSRBF.

in Fig. 4 are presented. In fact, in our approach, aggregating of effects from the adjacent elements by using Eq. (33) is similar to the mesh dependency filter in the work of Sigmund [51]. In Eq. (33), the radius rd has a fixed length scale that does not change with mesh refinement, which determines the number of concerned adjacent elements. Weight parameter w_{ij} indicates that the adjacent element has a greater effect on the central element when it is closer to the center. It is noticed that the operator in Eq. (33) smooths the shape gradient densities over the entire 3D design domain, and then the new shape gradient densities are applied in the optimization process to overcome the mesh dependency.
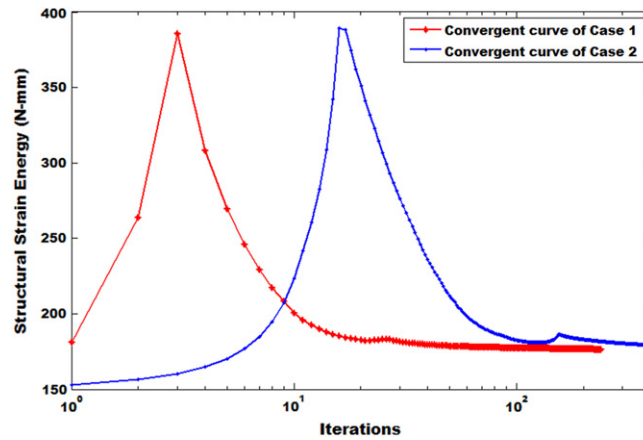
Fig. 6. Iteration histories for Case 1 and Case 2.

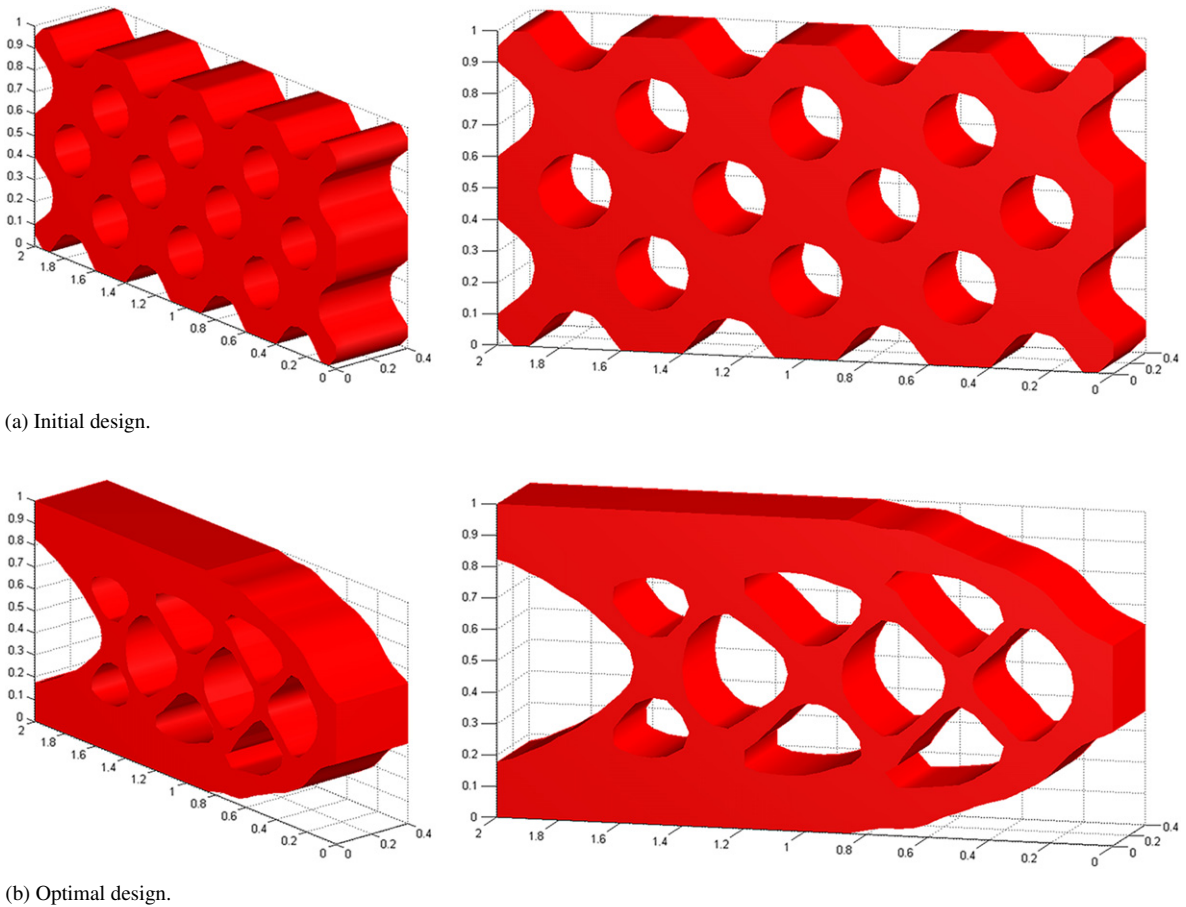

(a) Initial design.

(b) Optimal design.

Fig. 7. Initial design and optimal topologies for the single loading case considering extrusion constraint: (a) initial design; (b) optimal design.

The design domain is discretized with different meshes of $20 \times 4 \times 40$, $30 \times 6 \times 60$ and $40 \times 8 \times 80$. Accordingly, the grids for the evolving of level set functions are also $20 \times 4 \times 40$, $30 \times 6 \times 60$ and $40 \times 8 \times 80$. The extrusion path for each case is along **Y** axis. The radius rd for selecting the adjacent elements is set to 1.8 times of the mesh size. The optimal configurations are shown in Fig. 9. All the optimal designs are characterized with smooth boundaries and
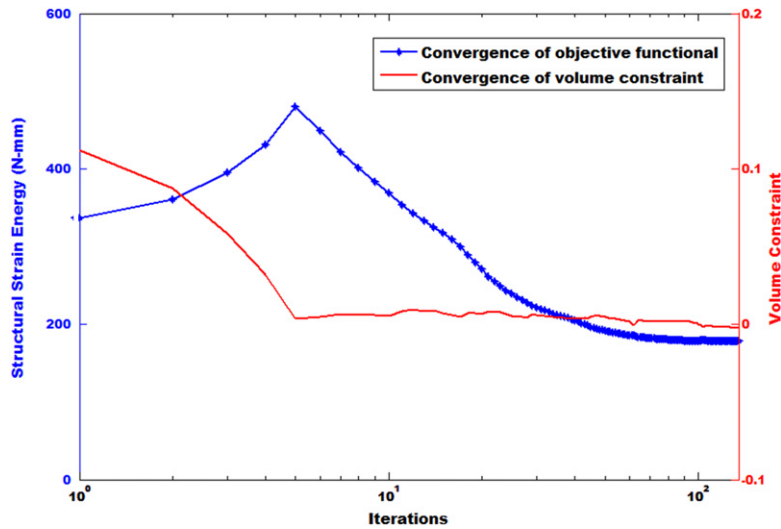
Fig. 8. Iteration histories for the single loading case considering extrusion constraint.

(a) Optimal design with mesh $20 \times 8 \times 40$.

(b) Optimal design with mesh $30 \times 12 \times 60$.

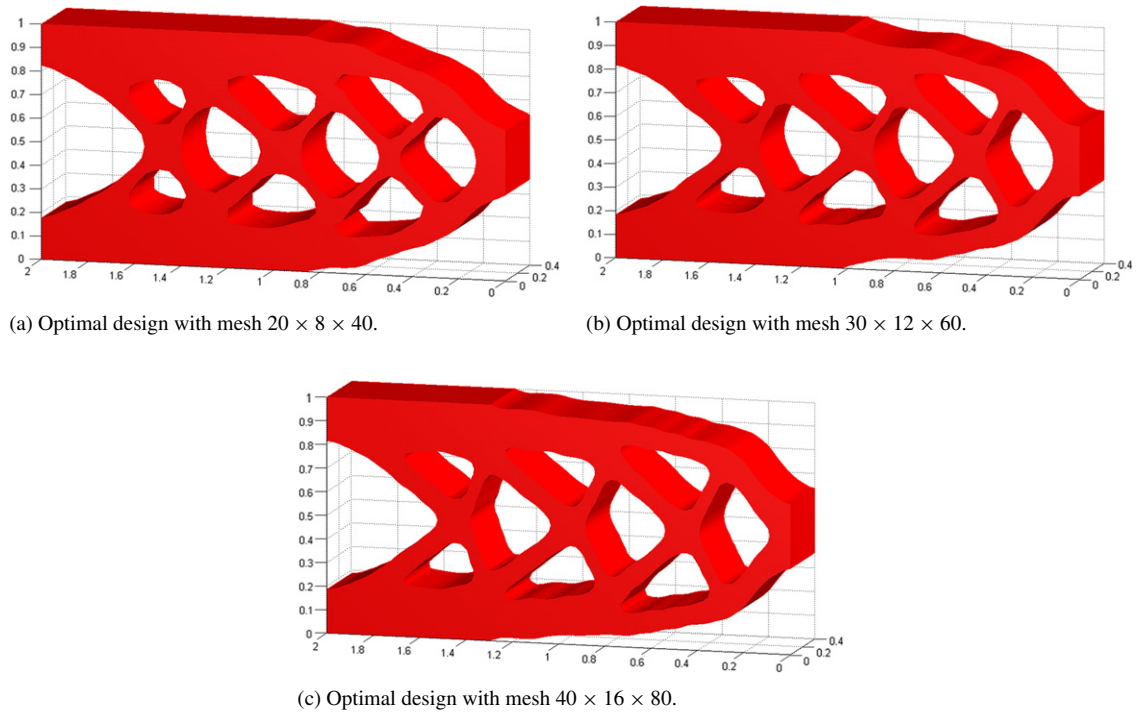(c) Optimal design with mesh $40 \times 16 \times 80$.

Fig. 9. Optimal configurations for different mesh sizes: (a) $20 \times 4 \times 40$; (b) $30 \times 6 \times 60$; (c) $40 \times 8 \times 80$.

constant cross sections. Although the shape of the case with relative coarser mesh (i.e. $20 \times 4 \times 40$) is slightly different from the other two cases, the structural topologies of the three cases are identical, which proves that the cross section projection strategy can effectively avoid mesh dependency.

5.2. Optimization problem with multiple loading cases

Now, we consider the compliance optimization problem under multiple loading cases. The design problem in 3D is shown in Fig. 10. Two horizontal edges of the bottom face of the structure are fixed as the Dirichlet boundary. Three
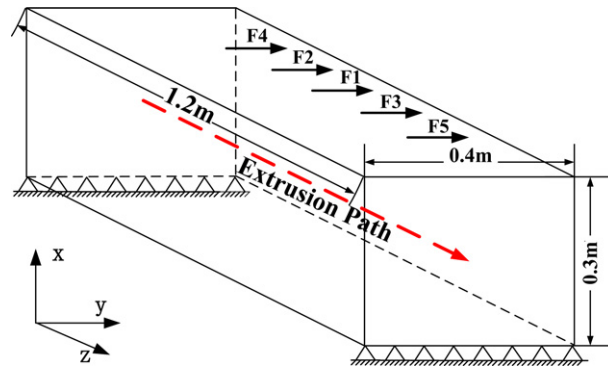
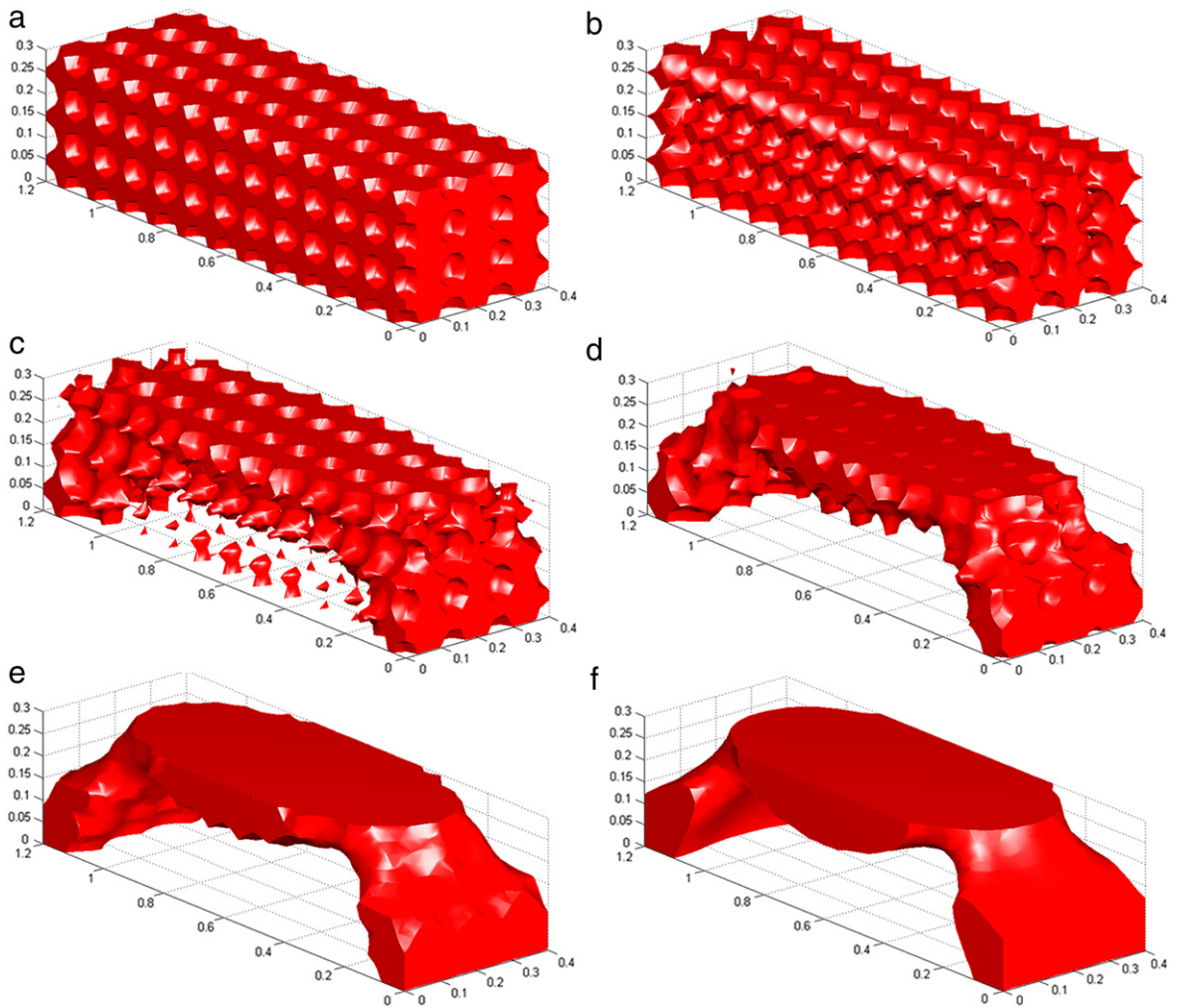Fig. 10. Design domain for the multiple loading cases.

Fig. 11. Structural topologies under multiple loading cases without extrusion constraint: (a) initial design; (b) intermediate design; (c) intermediate design; (d) intermediate design; (e) intermediate design; (f) optimal design.

loading cases are applied as the Neumann boundary, i.e. $F1 = 500\,\text{kN}$, $F2 = F3 = 300\,\text{kN}$ and $F4 = F5 = 200\,\text{kN}$ that are separately loaded at the top face. The objective is to optimize the compliance for each loading case under a constraint of 35% material usage.
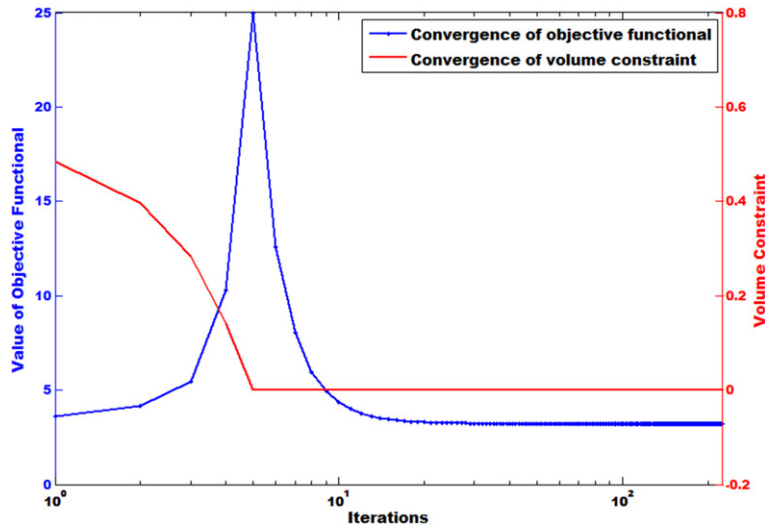
Fig. 12. Iteration histories for the optimization under multiple loading cases without extrusion constraint.

For a design problem considering multiple loading cases, it is actually a complicated multi-objective optimization problem, in which a Pareto solution is often included to represent a tradeoff between the ambivalent objectives. In this study, we implement the so-called exponential weighted criterion method to convert the multiple objectives into a single objective, which can capture the entire Pareto optimal set for the problems with both convex and non-convex Pareto frontier [52–54].

Thus, the alternative objective functional in Eq. (24) can be given as:

$$F(u, \Phi) = \sum_{k=1}^{m} (\exp(qw_k) - 1) \exp\left(q\left(\frac{J_k(u, \Phi) - J_k^{\min}}{J_k^{\max} - J_k^{\min}}\right)\right) \tag{43}$$

where F is the new objective functional derived from the original objective functional of each single loading case. w_k is the weight for each case and q is the exponential constant, which in this study is set as $q = 2$. $J_k^{\min}$ and $J_k^{\max}$ are utopia points for the kth loading case determined by solving the optimization problem of single loading case independently. m is the number of considered loading cases.

First, we examine the optimization problem without extrusion constraint. The linear elastic design domain with isotropic material is discretized with $12 \times 16 \times 48$ elements. We still implement the matrix compressing technique for the interpolant matrix $\mathbf{A}$. The factor $\kappa = 20$ is selected, and a much sparser $\mathbf{A}$ with 98.86% zeros is produced. Supposedly, the weight coefficients for the three loading cases are 0.5, 0.3 and 0.2, respectively. The utopia points for each load case are determined as $J_1^{\min} = 191.637$, $J_1^{\max} = 604.041$, $J_2^{\min} = 205.375$, $J_2^{\max} = 664.651$, $J_3^{\min} = 70.391$ and $J_3^{\max} = 238.358$, respectively. It should be remarked that the iteration is terminated when the difference of two successive objective values is less than 1×10^{-5} or when it has reached the maximum number of iterations, which is set to 400.

The initial design is plotted in Fig. 11(a), and the optimal structure is shown in Fig. 11(f). It is obvious that the entire structure freely enables shape variation and topological changes, by deleting existing holes and adding new holes within the 3D design domain. However, the optimal result is unable to be fabricated with the extrusion technique. Fig. 12 shows the convergent histories for the objective functional and volume constraint. It takes 225 iterations to optimize the value of objective functional from 24.97276 to 3.20205. Correspondingly, the optimal compliances are 203.773 for the first loading case, 211.317 for the second loading case and 77.407 for the third loading case. The CPU-time of the optimization is approximately 343.619 s per iteration.

Second, we discuss the extrusion-based design problem by using the cross section projection method. The design domain as well as the configuration of the optimization algorithm is the same as the first example in this section. The extrusion direction is along the $\mathbf{Z}$ axis. Nevertheless, the initial design is still different from that in the previous example, which can be seen in Fig. 13(a). The initial voids are with the same shape along $\mathbf{Z}$ axis. For the
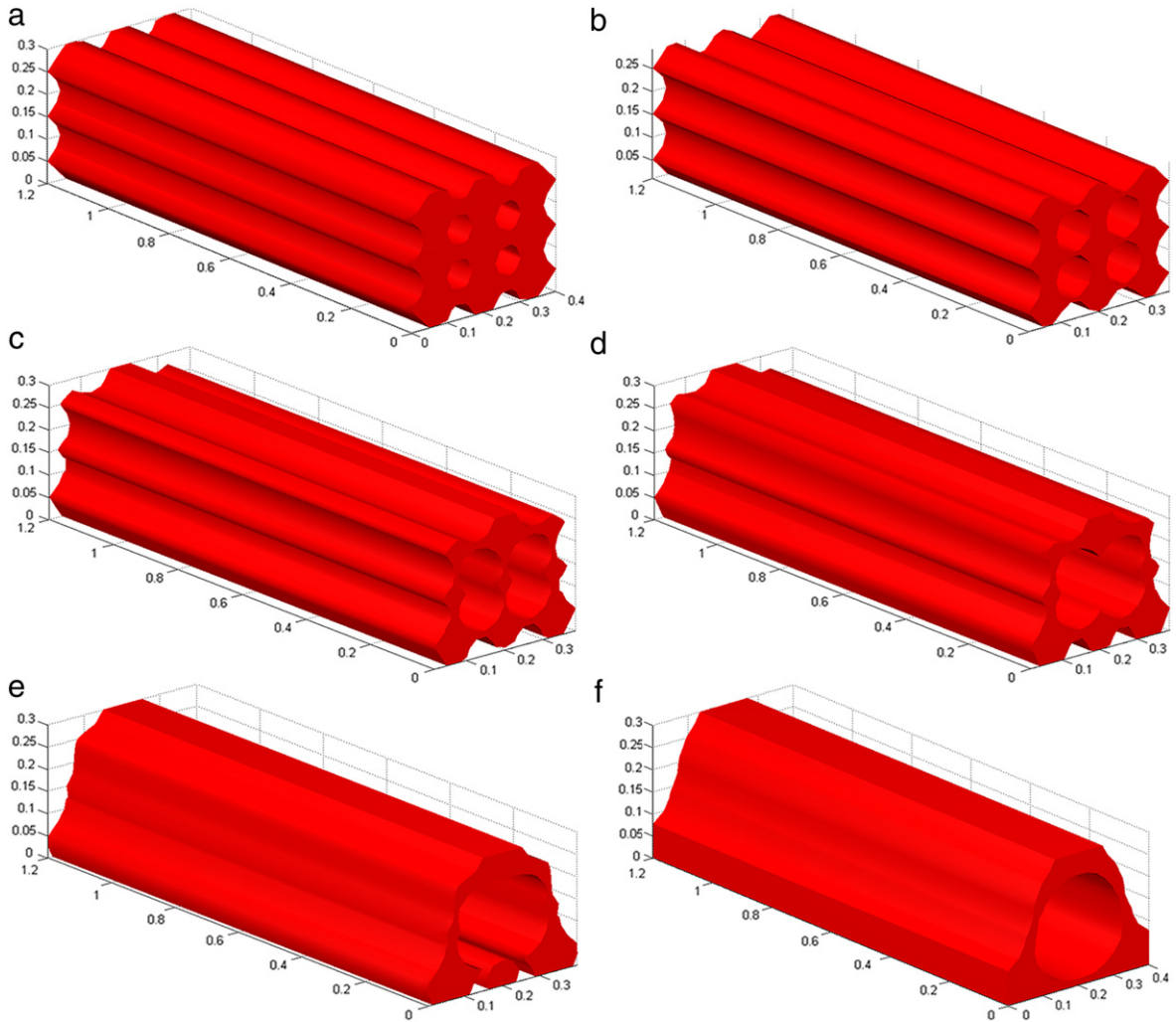
Fig. 13. Structural topologies under multiple loading cases considering extrusion constraint: (a) initial design; (b) intermediate design; (c) intermediate design; (d) intermediate design; (e) intermediate design; (f) optimal design.

extrusion-based optimization problem, the utopia points for each loading case are determined as $J_1^{\min} = 295.848$, $J_1^{\max} = 2721.224$, $J_2^{\min} = 318.718$, $J_2^{\max} = 3536.313$, $J_3^{\min} = 127.267$ and $J_3^{\max} = 1305.606$, respectively.

Fig. 13(b)–(f) show the evolving of structural topologies and Fig. 14 gives the corresponding cross-sections. One can notice that the cross section maintains smooth and constant for each step. In addition, this final result is of simple and identical geometry along the $\mathbf{Z}$ direction, and it can be manufacturing by extrusion along the pre-defined path.

Fig. 15 showcases the convergent histories about the objective and constraint. It takes 106 steps to minimize the value of objective functional from 26.65953 to 3.08652. In another aspect, the structural compliances are 314.907 for the first loading case, 352.112 for the second loading case and 138.934 for the third loading case. In addition, we observe the numerical stability and mass conservative. The computational time for single iteration is approximately 222.183 s, which is far less than that of the case without cross section projection method (343.619 s per step).

When comparing to the elemental density-based methods (such as SIMP et al.), the proposed level set-based parameterization method can produce a design that is more suitable for manufacturing in engineering. As we have known, the SIMP is a simple and convenient method for the topology optimization design. However, the optimized topologies may be difficult to extract due to the intermediate densities and zigzag geometries. On the contrary, the level set methods can always produce a smooth structural geometry. In addition, by introducing CSRBFs and DWT scheme, the interpolation efficiency for the parametric level set method has been dramatically improved. In fact, there is little
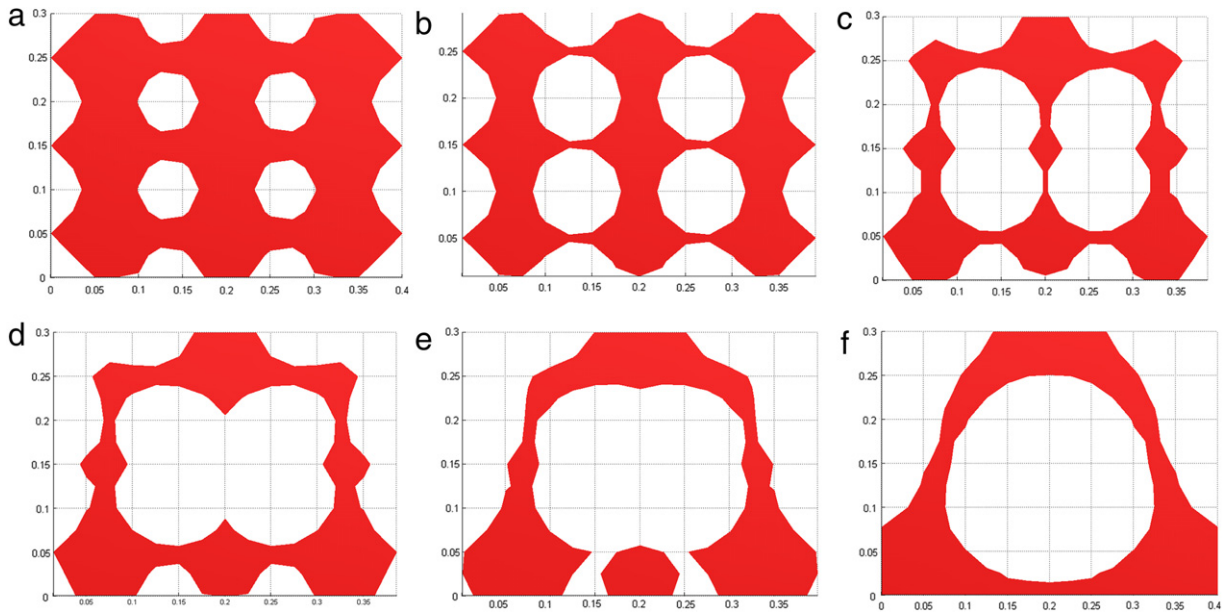
Fig. 14. Cross-sections corresponding to structural topologies under multiple loading cases: (a) initial design; (b) intermediate design; (c) intermediate design; (d) intermediate design; (e) intermediate design; (f) optimal design.
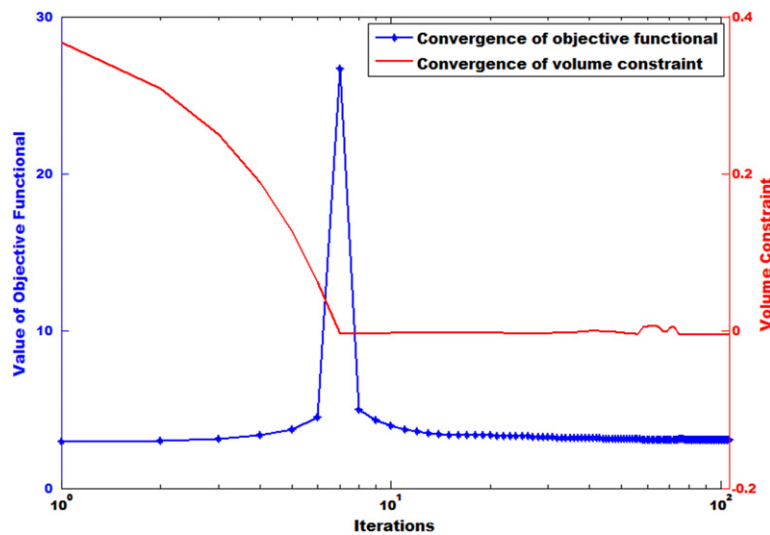

Fig. 15. Iteration histories for the optimization under multiple loading cases with extrusion constraint.

computational effort added into the optimization and no elaborated numerical operation is needed for the propagation of level set surface. That is to say, the main computational cost is the FEA, which makes the efficiency of proposed method comparable to that of the SIMP method. These facts are manifested in the aforementioned quantitative results.

6. Conclusions

This study has proposed an effective level set method for topological shape optimization of 3D structures considering the extrudable manufacturing constraint. To overcome the drawbacks and retain the unique advantages of level set methods, the CSRBFs are introduced to implement the interpolation of the level set function, and then the DWT is used to save the computational cost that arises from the application of the CSRBF via the sparsification of the coefficient matrix. In this way, there are a large number of zero elements in the system, through which the computer storage

as well as the CPU times are greatly reduced, especially for the large-scale 3D problems. Considering the extrusion constraint in the optimization, a cross section projection strategy is developed to map the data in the 3D domain onto a corresponding projection 2D plane perpendicular to the extrusion direction, thus converting the 3D extrusion design problem into a 2D cross-sectional optimization problem. Therefore, the cross sections are maintained constant, and the numbers of design variables as well as the computational effort are greatly reduced. Several numerical examples are used to showcase the characteristics of the proposed method. Our future work is to extend the proposed level set method to incorporate other manufacturing constraints, such as the member size constraint and the casting constraint.

Acknowledgments

References

[1] M.P. Bendsøe, O. Sigmund, Topology Optimization: Theory, Methods, and Applications, Springer, Berlin, Heidelberg, 2003.
[2] M.P. Bendsøe, N. Kikuchi, Generating optimal topology in structural design using a homogenization method, Comput. Methods Appl. Mech. Engrg. 71 (1988) 197–224.
[3] M. Zhou, G.I.N. Rozvany, The COC algorithm — Part II: topological, geometry and generalized shape optimization, Comput. Methods Appl. Mech. Engrg. 89 (1991) 197–224.
[4] M.P. Bendsøe, O. Sigmund, Material interpolation schemes in topology optimization, Arch. Appl. Mech. 69 (1999) 635–654.
[5] Z. Kang, Y.Q. Wang, Structural topology optimization based on non-local Shepard interpolation of density field, Comput. Methods Appl. Mech. Engrg. 200 (2011) 3515–3525.
[6] Z. Luo, N. Zhang, Y. Wang, W. Gao, Topology optimization of structures using meshless density variable approximants, Int. J. Numer. Methods Engrg. 93 (2013) 443–464.
[7] J.A. Sethian, A. Wiegmann, Structural boundary design via level set and immersed interface methods, J. Comput. Phys. 163 (2000) 489–528.
[8] M.Y. Wang, X.M. Wang, D.M. Guo, A level set method for structural topology optimization, Comput. Methods Appl. Mech. Engrg. 192 (2003) 227–246.
[9] G. Allaire, F. Jouve, A.M. Toader, Structural optimization using sensitivity analysis and a level-set method, J. Comput. Phys. 194 (2004) 363–393.
[10] Y.M. Xie, G.P. Steven, A simple evolutionary procedure for structural optimization, Comput. Struct. 49 (1993) 885–896.
[11] Y.Y. Kim, T.S. Kim, Topology optimization of beam cross sections, Int. J. Solids Struct. 37 (2000) 477–493.
[12] M. Zhou, R. Fleury, Y.K. Shyy, H. Thomas, J.M. Brennan, Progress in topology optimization with manufacturing constraints, in: 9th AIAA/ISSMO Symposium on Mutlidisciplinary Analysis and Optimization, AIAA, Atlanta, Georgia, 2002.
[13] K. Ishii, S. Aomura, Topology optimization for the extruded three dimensional structure with constant cross section, JSME Int. J. Ser. A-Solid Mech. Mater. Eng. 47 (2004) 198–206.
[14] S.T. Liu, X.M. An, H.P. Jia, Topology optimization of beam cross-section considering warping deformation, Struct. Multidiscip. Opt. 35 (2007) 403–411.
[15] N.M. Patel, C.L. Penninger, J.E. Renaud, Topology synthesis of extrusion-based nonlinear transient designs, ASME J. Mech. Des. 131 (2009) 061003.1–061003.11.
[16] R.H. Zuberi, Z.X. Zuo, K. Long, Topological optimization of beam cross section by employing extrusion constraint, in: Proceedings of ISCMII and EPMESCXII, AIP Publishing, Hong Kong-Macau, China, 2009, pp. 964–969.
[17] T. Yamada, K. Izui, S. Nishiwaki, A. Takezawa, A topology optimization method based on the level set method incorporating a fictitious interface energy, Comput. Methods Appl. Mech. Engrg. 199 (2010) 2876–2891.
[18] S. Osher, J.A. Sethian, Front propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 78 (1988) 12–49.
[19] J.A. Sethian, Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Version and Material Science, Cambridge University Press, UK, 1999.
[20] S. Osher, R.P. Fedkiw, Level Set Methods and Dynamic Implicit Surface, Springer, New York, 2002.
[21] M.Y. Wang, X.M. Wang, 'Color' level sets: a multiphase method for structural topology optimization with multiple materials, Comput. Methods Appl. Mech. Engrg. 193 (2004) 469–496.
[22] G. Allaire, F. Jouve, A level-set method for vibration and multiple loads structural optimization, Comput. Methods Appl. Mech. Engrg. 194 (2005) 3269–3290.
[23] S. Yamasaki, S. Nishiwaki, T. Yamada, K. Izui, M. Yoshimura, A structural optimization method based on the level set method using a new geometry-based re-initialization scheme, Int. J. Numer. Methods Engrg. 83 (2010) 1580–1624.
[24] Z. Luo, L.Y. Tong, M.Y. Wang, S.Y. Wang, Shape and topology optimization of compliant mechanisms using a parameterization level set method, J. Comput. Phys. 227 (2007) 680–705.
[25] N.P. van Dijk, K. Maute, M. Langelaar, F. van Keulen, Level-set methods for structural topology optimization: a review, Struct. Multidiscip. Opt. 48 (2013) 1–36.
[26] D. Makhija, K. Maute, Numerical instabilities in level set topology optimization with the extended finite element method, Struct. Multidiscip. Opt. 2013, http://dx.doi.org/10.1007/s00158-013-0982-x.

[27] T. Belytschko, S.P. Xiao, C. Parimi, Topology optimization with implicitly function and regularization, Int. J. Numer. Methods Engrg. 57 (2003) 1177–1196.
[28] E. Haber, A multilevel, level-set method for optimizing eigenvalues in shape design problems, J. Comput. Phys. 198 (2004) 518–534.
[29] Z. Luo, M.Y. Wang, S.Y. Wang, P. Wei, A level set-based parameterization method for structural shape and topology optimization, Int. J. Numer. Methods Engrg. 76 (2008) 1–26.
[30] J.Z. Luo, Z. Luo, L.P. Chen, L.Y. Tong, M.Y. Wang, A semi-implicit level set method for structural shape and topology optimization, J. Comput. Phys. 227 (2008) 5561–5581.
[31] S.Y. Wang, M.Y. Wang, Radial basis functions and level set method for structural topology optimization, Int. J. Numer. Methods Engrg. 65 (2006) 2060–2090.
[32] Z. Luo, N. Zhang, W. Gao, H. Ma, Structural shape and topology optimization using a meshless Galerkin level set method, Int. J. Numer. Methods Engrg. 90 (2012) 369–389.
[33] N.P. van Dijk, M. Langelaar, F. van Keulen, Explicit level-set-based topology optimization using an exact Heaviside function and consistent sensitivity analysis, Int. J. Numer. Methods Engrg. 91 (2012) 67–97.
[34] M. Mohamadian, S. Shojaee, Binary level set method for structural topology optimization with MBO type of projection, Int. J. Numer. Methods Engrg. 89 (2012) 658–670.
[35] J. Sokolowski, J.P. Zolesio, Introduction to Shape Optimization: Shape Sensitivity Analysis, Springer, Berlin, 1992.
[36] K.K. Choi, N.H. Kim, Structural Sensitivity Analysis and Optimization — Linear Systems, Springer, New York, 2005.
[37] M.D. Buhmann, Radial Basis Function: Theory and Implementations, in: Cambridge Monographs on Applied and Computational Mathematics, vol.12, Cambridge University Press, New York, 2004.
[38] C.E. Torres, L.A. Barba, Fast radial basis function interpolation with Gaussians by localization and iteration, J. Comput. Phys. 228 (2009) 4976–4999.
[39] S. Rippa, An algorithm for selecting a good value for the parameter c in radial basis functions interpolation, Adv. Comput. Math. 11 (1999) 193–210.
[40] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, Adv. Comput. Math. 4 (1995) 389–396.
[41] R. Schaback, H. Wendland, Inverse and saturation theorems for radial basis function interpolation, Math. Comput. 71 (2002) 669–681.
[42] C.H. Hsia, J.M. Guo, J.S. Chiang, A fast Discrete Wavelet Transform algorithm for visual processing applications, Signal Process. 92 (2012) 89–106.
[43] K. Chen, Discrete wavelet transforms accelerated sparse preconditioners for dense boundary element systems, Electron. Trans. Numer. Anal. 8 (1999) 138–153.
[44] J.M. Ford, E.E. Tyrtyshnikov, Combining Kronecker product approximation with discrete wavelet transforms to solve dense, function-related linear systems, SIAM J. Sci. Comput. 25 (2003) 961–981.
[45] P. González, J.C. Cabaleiro, T.F. Pena, Parallel iterative solvers involving fast wavelet transforms for the solution of BEM systems, Adv. Eng. Softw. 33 (2002) 417–426.
[46] I. Daubechies, Orthonormal bases of compactly supported wavelets, Comm. Pure Appl. Math. 41 (1988) 909–996.
[47] J. Ravnik, L. Škerget, M. Hriberšek, The wavelet transform for BEM computational fluid dynamics, Eng. Anal. Bound. Elem. 28 (2004) 1303–1314.
[48] G. Beylkin, R. Coifman, V. Rokhlin, Fast wavelet transforms and numerical algorithms, Comm. Pure Appl. Math. 44 (1991) 141–183.
[49] G.I.N. Rozvany, U. Kirsch, M.P. Bendsøe, O. Sigmund, Layout optimization of structures, Appl. Mech. Rev. 48 (1995) 41–119.
[50] K. Svanberg, The method of moving asymptotes: a new method for structural optimization, Int. J. Numer. Methods Engrg. 24 (1987) 359–373.
[51] O. Sigmund, J. Petersson, Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima, Struct. Optim. 16 (1998) 68–75.
[52] T.W. Athan, P.Y. Papalambros, A note on weighted criteria methods for compromise solutions in multi-objective optimization, Eng. Optim. 27 (1996) 155–176.
[53] H. Li, L. Gao, P.G. Li, Topology optimization of structures under multiple loading cases with a new compliance-volume product, Eng. Optimiz. 46 (2014) 725–744.
[54] R.T. Marler, J.S. Arora, Survey of multi-objective optimization methods for engineering, Struct. Multidiscip. Optmim. 26 (2004) 369–395.