

Massively efficient filter for topology optimization based on the splitting of tensor product structure

Aodi YANG, Shuting WANG, Nianmeng LUO, Tifan XIONG, Xianda XIE (✉)

School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

✉ Corresponding author. E-mail: xiandaxie@hust.edu.cn (Xianda XIE)

© Higher Education Press 2022

ABSTRACT In this work, we put forward a massively efficient filter for topology optimization (TO) utilizing the splitting of tensor product structure. With the aid of splitting technique, the traditional weight matrices of B-splines and non-uniform rational B-spline implicit filters are decomposed equivalently into two or three submatrices, by which the sensitivity analysis is reformulated for the nodal design variables without altering the optimization process. Afterwards, an explicit sensitivity filter, which is decomposed by the splitting pipeline as that applied to implicit filter, is established in terms of the tensor product of the axial distances between adjacent element centroids, and the corresponding sensitivity analysis is derived for elemental design variables. According to the numerical results, the average updating time for the design variables is accelerated by two-order-of-magnitude for the decomposed filter compared with the traditional filter. In addition, the memory burden and computing time of the weight matrix are decreased by six- and three-order-of-magnitude for the decomposed filter. Therefore, the proposed filter is massively improved by the splitting of tensor product structure and delivers a much more efficient way of solving TO problems in the frameworks of isogeometric analysis and finite element analysis.

KEYWORDS topology optimization, isogeometric analysis, finite element analysis, tensor product structure, sensitivity analysis

1 Introduction

Topology optimization (TO) refers to an innovative structural design tool in obtaining optimal material distribution under physical constraints [1]. With the continuous efforts of researchers in the TO community, several representative TO methods, including homogenization method [2], solid isotropic material with penalization (SIMP) model [3,4], evolutionary structural optimization (ESO) [5], level set method [6,7], and explicit geometric primitives approach [8–14], are emerging. Due to its algorithmic simplicity and effectiveness, the variable density methods, including SIMP and ESO, achieve their widespread popularity in solving TO design problems. Due to the 0 to 1 discreteness of the optimization model, the variable density methods inevitably encounter several numerical artifacts (e.g., mesh-dependency and checkerboard), if the additional treatment is not considered.

To overcome the aforementioned numerical issues, regularization techniques, which consist of imposing

additional restrictions on the solution space, such as perimeter constraints [15], gradient and slope constraints [16], and the filtering technique with the user-defined filter radius [17], are the most common used ones. Among these approaches, filtering technique is the most effective one and has become an essential ingredient for the variable density-based TO method, which evolves into sensitivity and density filters in an explicit form [18–21]. The key idea behind sensitivity and density filters is to replace the sensitivity of an arbitrary design variable with the average of sensitivities over its neighborhood region. Guest et al. [22] proposed linear and non-linear projection schemes based on regularized Heaviside step function using nodal variables to impose the minimum length scale constraint on the converged TO designs. Sigmund [23] considered the manufacturing tolerances of the optimized designs with image morphology-based filter. Xu et al. [24] presented a volume preserving Heaviside density filter to ensure the efficiency and stability of optimization. Kawamoto et al. [25] proposed a Heaviside projection scheme using a

Helmholtz partial differential equation (PDE)-based filter. Lazarov and Sigmund [26] obtained the isotropic and anisotropic density filters by solving a Helmholtz-type PDE with homogeneous Neumann boundary conditions. Chen et al. [27] applied the Heaviside filter to the TO of sound-absorbing materials in the framework of isogeometric boundary element method. Wallin et al. [28] added some penalty terms into an augmented PDE filter, which could tune the boundary properties using a scalar parameter. Xie et al. [29–31] proposed an adaptive filter with an adjusted radius under the adaptive isogeometric analysis (IGA) framework using hierarchical mesh.

However, the explicit filters mentioned above easily get into the trouble of the unbearable memory burden, even with very moderate filter radius for 3D problems. Qian [32] developed an implicit filter through the density field parameterized by B-splines and extended it to 3D design problems [33]. Costa et al. [34,35] imposed a local curvature constraint on the non-uniform rational B-splines (NURBS)-based implicit filter, which was used to implement the minimum and maximum length scale control [36,37], harmonic responses [38], structural displacement requirement [39,40], stress-constraint [41], multi-scale configuration [42,43], heat conduction [44], cellular materials [45], and anisotropic continua structure TO problems [46]. Fernandez et al. [47] regularized the TO model by B-spline design parameterization for the design problems of multiple contacted deformable bodies with large deformations. Wang et al. [48] developed a B-spline-based TO method under the overhang constraint that resulted from the additive manufacturing of the optimized designs, with the density gradient expressed explicitly.

Although the aforementioned implicit filters can resolve the storage problem by storing the univariate B-spline basis functions in all directions and the indices of basis function non-vanishing on each design element, it

ignores the side effect on efficiency that resulted from the additional iterative loops for determining the sensitivity of each nodal design variables from its supported regions, in comparison with the explicit filters, as mentioned above. The deficiency of implicit filters lies in the inconsistency between the data storing structure and the data processing structure in sensitivity analysis because the basis values on each design element are obtained from the tensor structure of B-splines. Tensor decomposition [49] corresponds to a technique applied to data arrays for extracting and expressing their properties, which has been applied to signal processing [50], deep learning [51], linear algebra [52], and IGA [53,54]. Inspired by this technique, we introduced a splitting technique for the tensor product structure-based TO implicit filter in this work, by which the matrix of B-splines values over all elements are decomposed into several smaller separated submatrices for different parametric directions. Each separated submatrix consists of the B-spline values over the elements along its associated parametric direction (Fig. 1). Afterwards, the sensitivity analysis is implemented by multiplying the elemental strain energy matrix with the separated submatrices in a sequential manner. The computation and storage complexity are reduced simultaneously from $O(n^2)$ to $O(n)$ for the TO filter, supposing that $n = ne_x = ne_y$, by means of the procedure mentioned above. Then, we extend this splitting technique to the TO sensitivity filter in the explicit form, where the weights defined by the centroid distances are replaced by the tensor product of the centroid distances along all axis directions. Therefore, the numerical and storage efficiencies are improved simultaneously, and the conflict between storage burden and computational efficiency is largely alleviated for the sensitivity filter of the variable density methods.

In the remainder of this work, Section 2 describes the implementation details for the tensor product splitting

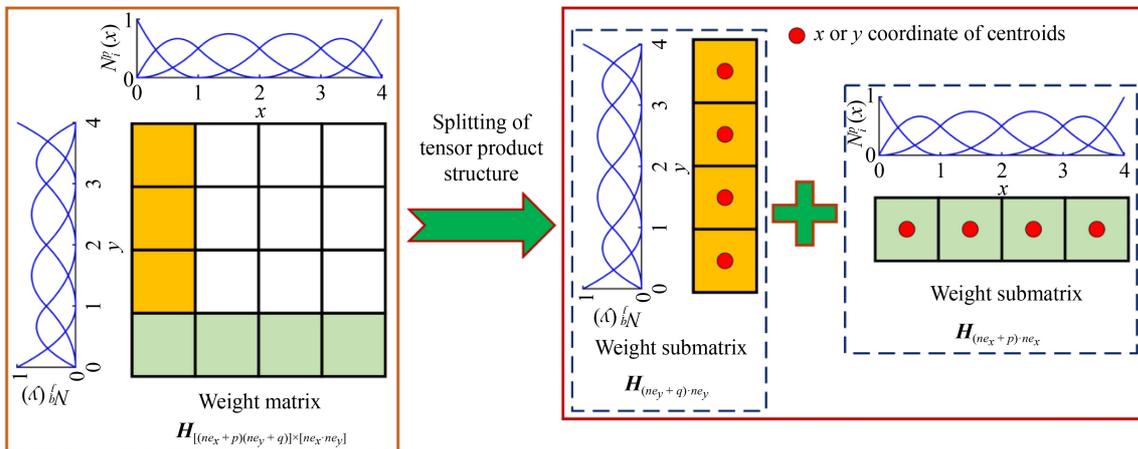


Fig. 1 Illustration of the decomposition of the weight matrix into two weight submatrices for B-splines implicit filter, with ne_x and ne_y representing the number of elements along with the x and y directions, respectively, and p and q denoting the degrees of B-splines in the x and y directions, respectively.

technique-based TO method. Section 3 illustrates the construction and splitting processes for the explicit sensitivity filter in a tensor product form. Section 4 validates the effectiveness of the splitting technique for the TO implicit filter in IGA-based TO framework. Section 5 verifies the performance of tensor product-based explicit sensitivity filter combined with the proposed splitting technique in finite element method (FEM)-based TO framework. Finally, the conclusions are drawn at the end of this paper.

2 TO based on the splitting of implicit B-splines filter

In this section, the SIMP-based TO model is first reviewed for minimization compliance problems in the IGA framework with the implicit B-spline filters. Then, the sensitivity analysis is reformulated in terms of the splitting technique of B-splines tensor product structure, and the data structure is proposed for the decomposed implicit B-spline filters. The computing and storing complexities between the original implicit B-spline filters and the decomposed one are analyzed and compared.

2.1 TO using non-uniform B-spline parametrization

On the top of the TO model with the density field parametrized by uniform B-splines [32], we parametrize the density design field by non-uniform B-splines to ensure consistency between the design domain and the parametrized design field (Fig. 2). The nodal densities

associated with the control points of B-splines are treated as the coefficients of the density parameterization scheme, which is formulated as

$$\rho(x, y) = \sum_{i=1}^m \sum_{j=1}^n N_{i,j}^{p,q}(x, y) \rho_{i,j}, \quad (1)$$

where (x, y) denotes the coordinates of an arbitrary point in the design domain with the density $\rho(x, y)$, m and n represent the number of control points along with the horizontal and vertical directions, respectively, $\rho_{i,j}$ is the nodal density of the (i, j) th control points, and the bivariate B-spline basis function $N_{i,j}^{p,q}(x, y)$ with $(p \times q)$ -degree combinations is obtained from the tensor product structure of B-splines and in the form of

$$N_{i,j}^{p,q}(x, y) = N_i^p(x) \cdot N_j^q(y), \quad (2)$$

where $N_i^p(x)$ is the i th B-spline basis function along with the x direction with degree equals to p and $N_j^q(y)$ denotes the j th B-spline basis function along with the y direction with degree equals to q , which are defined by Cox-de Boor recursion formula [55] as follows:

$$N_i^p(x) = \frac{x - x_i}{x_{i+p} - x_i} N_i^{p-1}(x) + \frac{x_{i+1} - x}{x_{i+1} - x_i} N_{i+1}^{p-1}(x), \quad (3)$$

with $N_i^0(x) = \begin{cases} 1, & x_i \leq x < x_{i+1}, \\ 0, & \text{otherwise,} \end{cases}$

where x is the knot coordinate for the $(p + 1)$ order non-decreasing knot vector $\Xi = \{x_0, x_1, \dots, x_{n+p+1}\}$, which belongs to the i th interval $[x_i, x_{i+1}]$.

Therefore, the TO model using non-uniform B-spline parametrization under the Dirichlet boundary is formulated as

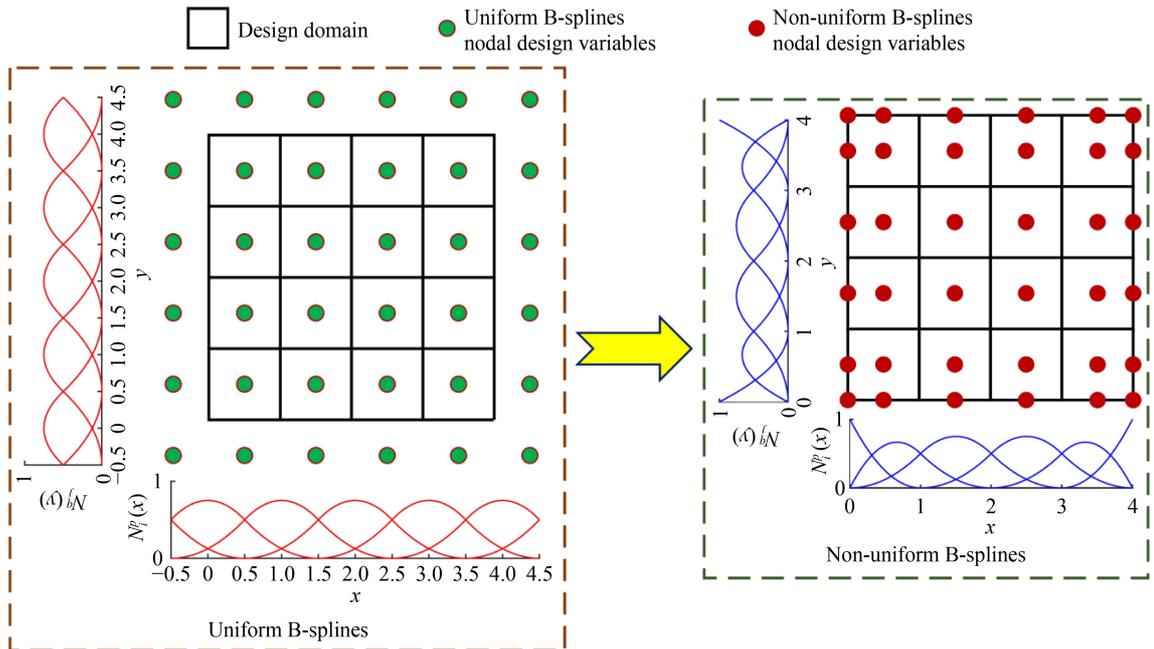


Fig. 2 Illustration of the substitution of non-uniform B-splines for uniform B-splines as the nodal design variables for topology optimization.

$$\begin{aligned}
& \text{find } \boldsymbol{\rho} = (\rho_{1,1}, \rho_{1,2}, \dots, \rho_{i,j}, \dots, \rho_{m,n}), \\
& i = 1, 2, \dots, m, j = 1, 2, \dots, n, \\
& \text{minimize } c(\boldsymbol{\rho}) = \mathbf{f}^T \cdot \mathbf{u}(\boldsymbol{\rho}) = \sum_{ex=1}^{ne_x} \sum_{ey=1}^{ne_y} [E(\rho_{ex,ey}) \\
& \quad \cdot (\mathbf{u}_{ex,ey})^T \cdot \mathbf{k}_{ex,ey} \cdot \mathbf{u}_{ex,ey}], \\
& \mathbf{K}(\boldsymbol{\rho}) \cdot \mathbf{u}(\boldsymbol{\rho}) = \left(\sum_{ex=1}^{ne_x} \sum_{ey=1}^{ne_y} E(\rho_{ex,ey}) \cdot \mathbf{k}_{ex,ey} \right) \cdot \mathbf{u}(\boldsymbol{\rho}) = \mathbf{f}, \\
& \text{s.t. } \frac{V}{V_0} \leq \text{volfrac}, \\
& \boldsymbol{\rho} \subset \boldsymbol{\psi}, \boldsymbol{\psi} = \{\boldsymbol{\rho} \in \mathbb{R}^{m \times n}, 0 \leq \boldsymbol{\rho} \leq 1\},
\end{aligned} \tag{4}$$

where $c(\boldsymbol{\rho})$ is the compliance that measures the structural strain energy under the external load \mathbf{f} , $\mathbf{K}(\boldsymbol{\rho})$ and $\mathbf{u}(\boldsymbol{\rho})$ represent the global structural stiffness and displacement vector of control points over the design domain, respectively, $\mathbf{k}_{ex,ey}$ and $\mathbf{u}_{ex,ey}$ are the solid elemental structural stiffness and local displacement vector of control points for the (ex, ey) th IGA element, respectively, V and volfrac denote the volume and volume fraction of solid material, respectively, V_0 represents the volume of the whole design domain, $\boldsymbol{\psi}$ is the admissible space for the union of nodal design variables $\boldsymbol{\rho}$, and $E(\rho_{ex}, \rho_{ey})$ is termed as the interpreted Young's modulus by the modified SIMP model of isotropic material for the (ex, ey) th IGA element and written in:

$$E(\rho_{ex,ey}) = E_{\min} + (\rho_{ex,ey}^c)^{\text{penal}} \cdot (E_0 - E_{\min}), \tag{5}$$

where E_{\min} and E_0 denote the Young's modulus of void and solid elements with $0 < E_{\min} \ll E_0$, respectively, penal is the penalty factor to steer black and white design and takes 3 in this paper, and $\rho_{ex,ey}^c$ is the centroid density values for the (ex, ey) th IGA element and is obtained from Eq. (1), as follows:

$$\rho_{ex,ey}^c = \sum_{i=1}^m \sum_{j=1}^n N_{i,j}^{p,q}(x_{ex}^c, y_{ey}^c) \rho_{i,j}, \tag{6}$$

where (x_{ex}^c, y_{ey}^c) denotes the centroid coordinate of the (ex, ey) th IGA element.

2.2 Sensitivity analysis by the splitting of B-splines tensor product structure

Due to the local support property of B-splines (Fig. 3), the sensitivity of the objective function of Eq. (4) with respect to arbitrary nodal design variables $\rho_{i,j}$ can be

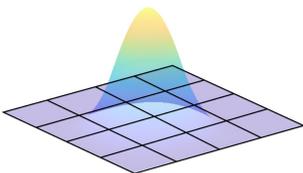


Fig. 3 Illustration of the local support property of B-spline basis function.

obtained by applying the differential chain rule over its supported design elements, which is formulated as

$$\frac{\partial c}{\partial \rho_{i,j}} = \sum_{ex=\max(1,i-p)}^{\min(i,ne_x)} \sum_{ey=\max(1,j-q)}^{\min(j,ne_y)} \frac{\partial c}{\partial \rho_{ex,ey}^c} \cdot \frac{\partial \rho_{ex,ey}^c}{\partial \rho_{i,j}}. \tag{7}$$

According to Eq. (5), $\frac{\partial c}{\partial \rho_{ex,ey}^c}$ is defined as

$$\begin{aligned} \frac{\partial c}{\partial \rho_{ex,ey}^c} &= -\text{penal} \cdot (\rho_{ex,ey}^c)^{\text{penal}-1} \cdot (E_0 - E_{\min}) \\ &\quad \cdot (\mathbf{u}_{ex,ey})^T \cdot \mathbf{k}_{ex,ey} \cdot \mathbf{u}_{ex,ey}, \end{aligned} \tag{8}$$

which is abbreviated as $dc_{ex,ey}$. Substituting Eq. (6) into Eq. (7), Eq. (7) is reformulated as

$$\frac{\partial c}{\partial \rho_{i,j}} = \sum_{ex=\max(1,i-p)}^{\min(i,ne_x)} \sum_{ey=\max(1,j-q)}^{\min(j,ne_y)} dc_{ex,ey} \cdot N_{i,j}^{p,q}(x_{ex}^c, y_{ey}^c). \tag{9}$$

Contrary to calculating the B-spline basis function presented in Eq. (9) in its tensor product form, we split the B-spline basis function of Eq. (9) into its two univariate B-spline basis functions. Then, Eq. (9) can be written as

$$\frac{\partial c}{\partial \rho_{i,j}} = \sum_{ex=\max(1,i-p)}^{\min(i,ne_x)} N_i^p(x_{ex}^c) \sum_{ey=\max(1,j-q)}^{\min(j,ne_y)} dc_{ex,ey} \cdot N_j^q(y_{ey}^c). \tag{10}$$

If $p < i \leq ne_x$ and $q < j \leq ne_y$, then Eq. (10) can be rewritten in its matrix form as follows:

$$\begin{aligned} \frac{\partial c}{\partial \rho_{i,j}} &= \left[N_i^p(x_{i-p}^c) \quad N_i^p(x_{i-p+1}^c) \quad \cdots \quad N_i^p(x_i^c) \right] \\ &\quad \cdot \begin{bmatrix} dc_{i-p,j-q} & dc_{i-p,j-q+1} & \cdots & dc_{i-p,j} \\ dc_{i-p+1,j-q} & dc_{i-p+1,j-q+1} & \cdots & dc_{i-p+1,j} \\ \vdots & \vdots & \ddots & \vdots \\ dc_{i,j-q} & dc_{i,j-q+1} & \cdots & dc_{i,j} \end{bmatrix} \cdot \begin{bmatrix} N_j^p(y_{j-q}^c) \\ N_j^p(y_{j-q+1}^c) \\ \vdots \\ N_j^p(y_j^c) \end{bmatrix}. \end{aligned} \tag{11}$$

Therefore, the sensitivities of the objective function with respect to all nodal design variables can be presented as

$$\begin{bmatrix} \frac{\partial c}{\partial \rho_{1,1}} & \frac{\partial c}{\partial \rho_{1,2}} & \cdots & \frac{\partial c}{\partial \rho_{1,n}} \\ \frac{\partial c}{\partial \rho_{2,1}} & \frac{\partial c}{\partial \rho_{2,2}} & \cdots & \frac{\partial c}{\partial \rho_{2,n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial c}{\partial \rho_{m,1}} & \frac{\partial c}{\partial \rho_{m,2}} & \cdots & \frac{\partial c}{\partial \rho_{m,n}} \end{bmatrix} = \mathbf{H}_{m \times ne_x}(x^c) \cdot \mathbf{dc}_{ne_x \times ne_y} \cdot \mathbf{H}_{ne_y \times n}(y^c), \tag{12}$$

where $\mathbf{H}_{m \times ne_x}(x^c)$ and $\mathbf{H}_{ne_y \times n}(y^c)$ denote the unions of univariate B-spline basis functions at the centroids of all design elements along with the x and y directions (Fig. 1), which are diagonal sparse matrices. $\mathbf{dc}_{ne_x \times ne_y}$ is the union of sensitivities of the objective function with respect to

the centroid densities for all design elements and is organized as

$$\mathbf{dc}_{ne_x \times ne_y} = \begin{bmatrix} dc_{1,1} & dc_{1,2} & \cdots & dc_{1,ne_y} \\ dc_{2,1} & dc_{2,2} & \cdots & dc_{2,ne_y} \\ \vdots & \vdots & \ddots & \vdots \\ dc_{ne_x,1} & dc_{ne_x,2} & \cdots & dc_{ne_x,ne_y} \end{bmatrix}_{ne_x \times ne_y}. \quad (13)$$

Similarly, the sensitivities of the volume constraint function associated with the nodal design variables are formulated as

$$\begin{bmatrix} \frac{\partial V}{\rho_{1,1}} & \frac{\partial V}{\rho_{1,2}} & \cdots & \frac{\partial V}{\rho_{1,n}} \\ \frac{\partial V}{\rho_{2,1}} & \frac{\partial V}{\rho_{2,2}} & \cdots & \frac{\partial V}{\rho_{2,n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial V}{\rho_{m,1}} & \frac{\partial V}{\rho_{m,2}} & \cdots & \frac{\partial V}{\rho_{m,n}} \end{bmatrix} = \mathbf{H}_{m \times ne_x}(x^c) \cdot \mathbf{dV}_{ne_x \times ne_y} \cdot \mathbf{H}_{ne_y \times n}(y^c), \quad (14)$$

with

$$\mathbf{dV}_{ne_x \times ne_y} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}, \quad (15)$$

where $\mathbf{dV}_{ne_x \times ne_y}$ is the union of sensitivities of the volume fraction with respect to the centroid densities for all elements. Furthermore, Eq. (14) can be reformulated as

$$\begin{bmatrix} \frac{\partial V}{\rho_{1,1}} & \frac{\partial V}{\rho_{1,2}} & \cdots & \frac{\partial V}{\rho_{1,n}} \\ \frac{\partial V}{\rho_{2,1}} & \frac{\partial V}{\rho_{2,2}} & \cdots & \frac{\partial V}{\rho_{2,n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial V}{\rho_{m,1}} & \frac{\partial V}{\rho_{m,2}} & \cdots & \frac{\partial V}{\rho_{m,n}} \end{bmatrix} = \begin{bmatrix} N_1^p(x^c) \\ \vdots \\ \sum_{ex=i-p}^i N_i^p(x^c) \\ \vdots \\ N_m^p(x^c_{ne_x}) \end{bmatrix} \cdot \begin{bmatrix} N_1^q(y^c_1) & \cdots & \sum_{ey=j-q}^j N_j^q(y^c_{ey}) & \cdots & N_n^q(y^c_{ne_y}) \end{bmatrix}. \quad (16)$$

Finally, by the splitting of tensor product structure, as presented in Eq. (11), we can obtain the sensitivity analysis of TO by the univariate B-spline basis function values in all parametric directions, rather than calculating the bivariate B-spline basis values in advance. In addition, the splitting technique can be easily extended from 2D case to 3D case, and two choices of the splitting process for tri-variate B-splines are depicted in Fig. 4.

The differences between the directional splitting and slice splitting are categorized into two aspects: On the one hand, the memory burden of directional splitting is much

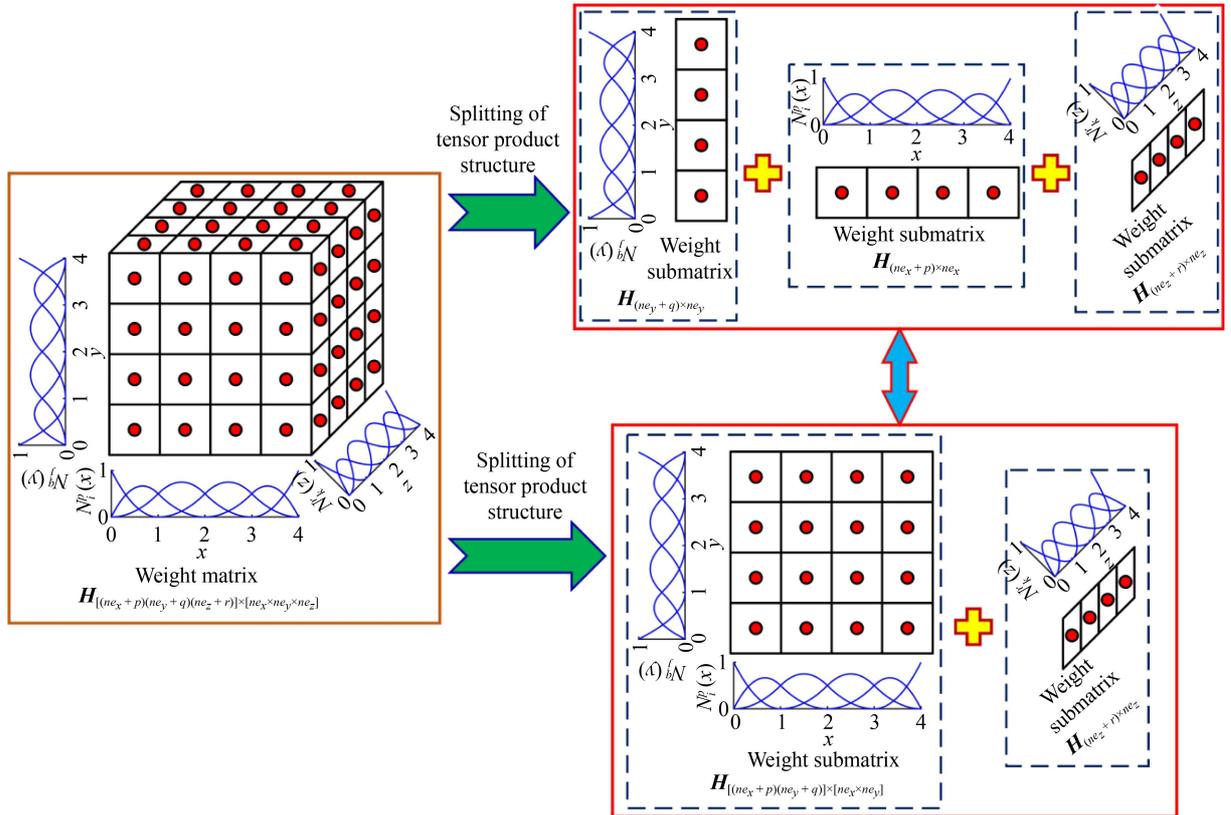


Fig. 4 Illustration of two options of the splitting process for the tensor product structure of tri-variate B-splines in 3D topology optimization implicit filter: directional splitting (on the right-top corner) and slice splitting (on the right-bottom corner).

lower than the slice splitting because the space complexity of the weight matrices is reduced from $O(n^4)$ to $O(n^2)$, as shown in Fig. 4, supposing that $ne_x = ne_y = ne_z = n$ with ne_z representing the number of elements along with the z direction; on the other hand, the updating time involved in sensitivity analysis for the directional splitting is slightly larger than slice splitting because of the overhead related to the matrix transformation with respect to $dc_{ne_x \times ne_y \times ne_z}$.

2.3 Computing and storing of the decomposed weight matrices of implicit filter

To calculate the decomposed B-spline weight matrices (Fig. 1), we need to determine the x -coordinates of the centroids of the light green elements and the y -coordinates of centroids of the light brown elements. Then, the B-spline basis values on a design element are determined by substituting its x -coordinate into Eq. (3) for the B-spline basis functions, which are filled into $H_{(ne_x+p) \times ne_x}$. In comparison with the efficient filtering technique by Wang and Qian [33], the computation of the B-spline weight factors is much cheaper for the decomposed B-spline weight matrices, as there is not necessary to calculate the B-spline weight factors of all axial directions by the tensor product structure for all design elements. The comparisons in computing time between the B-spline implicit filter in Ref. [33] and our pipelines are presented in Fig. 5 under three different degree combinations and five different mesh resolutions

for the pre-computed B-spline weight factors.

For the storing of the decomposed B-spline weight matrices, the cost of $H_{(ne_x+p) \times ne_x}$ and $H_{(ne_y+q) \times ne_y}$ are $S_x = 16 \cdot ne_x \cdot (p+1)$ and $S_y = 16 \cdot ne_y \cdot (q+1)$ Bytes (B), respectively. Consequently, the overall storing burden of B-spline weights is $S = 16 \cdot [ne_x \cdot (p+1) + ne_y \cdot (q+1)]$ B. Moreover, the overall storing burden of the B-spline weights of the directional and slice splitting are $S = 16 \cdot [ne_x \cdot (p+1) + ne_y \cdot (q+1) + ne_z \cdot (r+1)]$ B and $S = 16 \cdot [ne_x \cdot ne_y \cdot (p+1)(q+1) + ne_z \cdot (r+1)]$ B, respectively. The storage comparisons between the B-spline implicit filter in Ref. [33] and our proposed ones are presented in Fig. 6 for 3D problems under three different degree combinations and five different mesh resolutions.

3 TO based on explicit tensor product filter

In the variable density-based TO method, sensitivity or density filter is widely used in an explicit form, of which the weight matrix is defined by the centroid distance between adjacent design elements and usually suffers from the prohibitive memory burden for the fine design mesh. To resolve this issue explicitly, we extend the splitting technique from the implicit filter presented in Section 2 to the sensitivity-based explicit filter. In this section, we first review the TO model based on the variable density method in the FEM framework. Then, the construction of explicit filter by the tensor product structure is presented, where the weight defined by

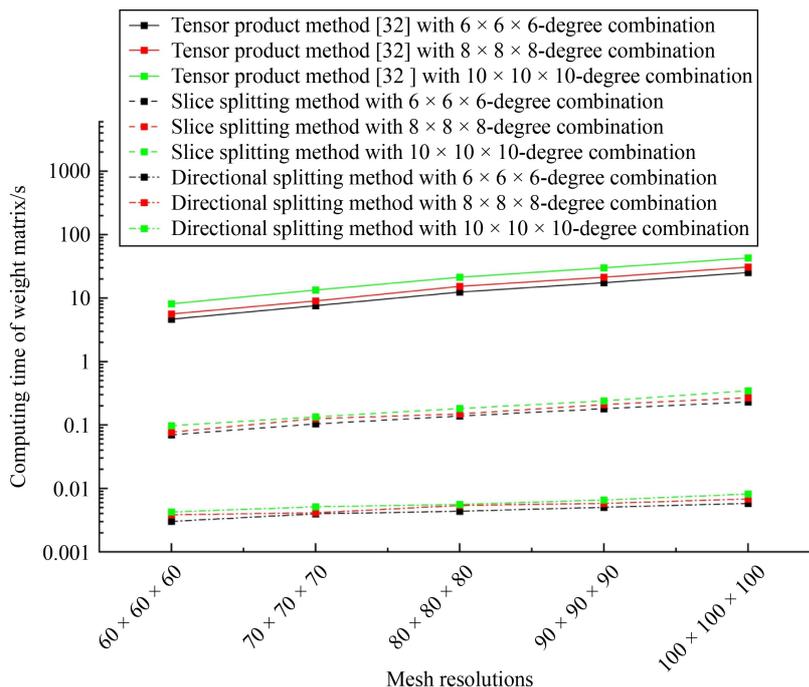


Fig. 5 Illustration of the comparisons in computing time for the pre-computed weight matrices among three different filtering methods for 3D problems.

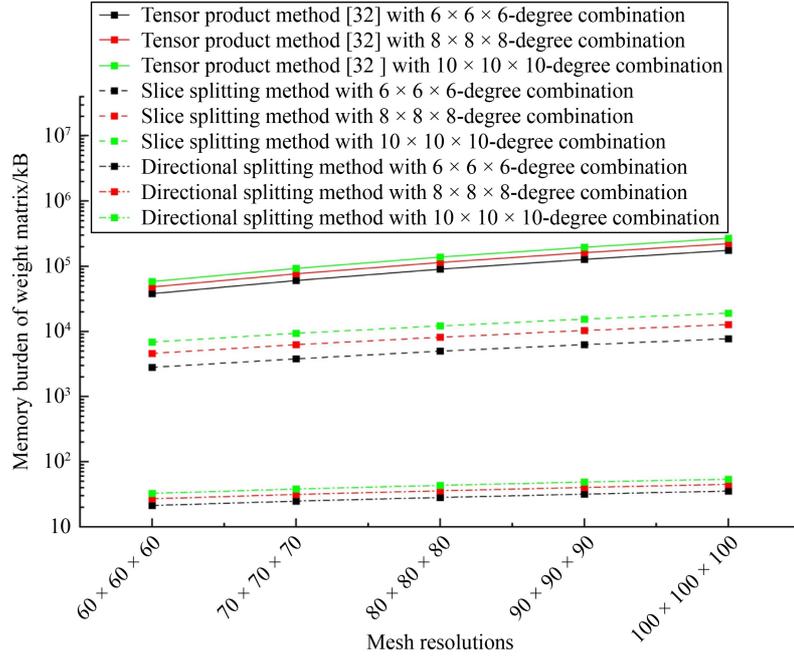


Fig. 6 Illustration of the comparison results in storage burden related to the weight matrix of B-spline implicit filter between the traditional one and the slice splitting one, as well as the directional splitting one for 3D topology optimization problems.

centroid distance is replaced by the tensor product of the centroid distances along with the axial directions. Afterwards, similar to the sensitivity analysis of the decomposed B-spline implicit filter, the sensitivity analysis for the decomposed explicit filter is derived on the basis of the splitting technique for 2D compliance TO problems.

3.1 TO using variable density method

In contrast to parametrizing the density field over the design domain by the nodal design variables, as presented in Section 2, the variable density method presented in this work describes the density field by the element-wise constant design variables in a discrete form. Therefore, the TO model that uses the density variable method for compliance problems is formulated as

$$\text{find } \bar{\rho} = (\bar{\rho}_{1,1}, \bar{\rho}_{1,2}, \dots, \bar{\rho}_{i,j}, \dots, \bar{\rho}_{ne_x, ne_y}),$$

$$i = 1, 2, \dots, ne_x, j = 1, 2, \dots, ne_y,$$

$$\text{minimize } c(\bar{\rho}) = \mathbf{f}^T \cdot \mathbf{u}(\bar{\rho}) = \sum_{i=1}^{ne_x} \sum_{j=1}^{ne_y} E(\bar{\rho}_{i,j}) \cdot (\mathbf{u}_{i,j})^T \cdot \mathbf{k}_0 \cdot \mathbf{u}_{i,j},$$

$$\mathbf{K}(\bar{\rho}) \cdot \mathbf{u}(\bar{\rho}) = \left(\sum_{i=1}^{ne_x} \sum_{j=1}^{ne_y} E(\bar{\rho}_{i,j}) \cdot \mathbf{k}_0 \right) \cdot \mathbf{u}(\bar{\rho}) = \mathbf{f},$$

$$\text{s.t. } \frac{V}{V_0} \leq \text{volfrac},$$

$$\bar{\rho} \subset \bar{\psi}, \bar{\psi} = \{\bar{\rho} \in \mathbb{R}^{ne_x \times ne_y}, 0 \leq \bar{\rho} \leq 1\}, \quad (17)$$

where $\bar{\rho}$ is the union of all discrete design variables in FEM, $\mathbf{u}(\bar{\rho})$ is termed as the global displacement vector of

the discretized FEM mesh, $\mathbf{u}_{i,j}$ is the local displacement vector for the (i,j) th finite element, \mathbf{k}_0 represents the standard elemental stiffness matrix for the finite element mesh, $\mathbf{K}(\bar{\rho})$ is termed as the global stiffness of the FEM mesh, $\bar{\psi}$ is the admissible space for $\bar{\rho}$, and the Young's elastic modulus $E(\bar{\rho}_{i,j})$ is defined by the modified SIMP method of isotropic material [19] for the (i,j) th finite element and formulated as

$$E(\bar{\rho}_{i,j}) = E_{\min} + (\bar{\rho}_{i,j})^{\text{penal}} \cdot (E_0 - E_{\min}). \quad (18)$$

3.2 Construction of explicit filter based on tensor product structure

To overcome the numerical artifacts that resulted from the discrete nature of design variables (e.g., mesh dependency and checkerboard pattern), variable density-based TO method usually treats the modified sensitivities of the design variables as the input parameters for optimality criterion [4] or the method moving asymptotes [56] optimizer. Given that the weight matrix for the modified sensitivities is defined by the centroid distance between adjacent elements (Fig. 7(a)), neither the sensitivity filter nor the density filter can utilize the splitting technique presented in Subsection 2.2 directly. To implement the splitting of explicit filter, we first construct an explicit sensitivity filter in terms of the tensor product structure of the centroid distances along the x and y directions between two adjacent elements (Fig. 7(b)), within the filtering regions defined by a specified filter radius.

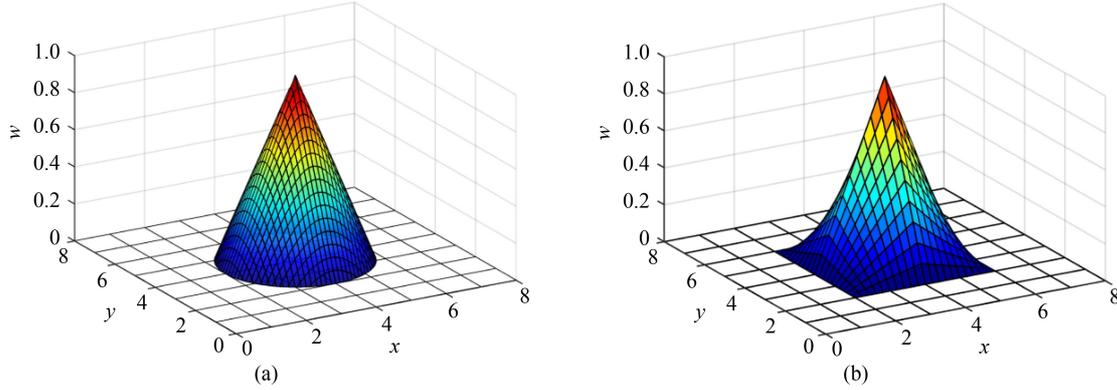


Fig. 7 Illustration of explicit sensitivity filters with $r_{\min} = 2$ for (a) the traditional centroid distance-based filter and (b) the tensor product structure-based filter.

Therefore, the explicit sensitivity filter based on the tensor product structure is defined as

$$w_{(i,j),(i_1,j_1)} = \frac{w_{i,i_1}(x) \cdot w_{j,j_1}(y)}{\sum_{(i_2,j_2) \in \mathbb{N}_{(i,j)}} w_{i,i_2}(x) \cdot w_{j,j_2}(y)}, \quad (19)$$

with

$$w_{i,i_1}(x) = \frac{\max(\lceil r_{\min} \rceil - |i - i_1|, 0)}{\lceil r_{\min} \rceil},$$

$$w_{j,j_1}(y) = \frac{\max(\lceil r_{\min} \rceil - |j - j_1|, 0)}{\lceil r_{\min} \rceil}, \quad (20)$$

where $w_{i,i_1}(x)$ and $w_{j,j_1}(y)$ represent the weights of the i_1 th and j_1 th design elements along with the x and y directions contributed to the modified sensitivity, respectively, $\mathbb{N}_{(i,j)}$ represents the neighborhood region of the (i,j) th element defined by the filter radius $\lceil r_{\min} \rceil$, and $w_{(i,j),(i_1,j_1)}$ denotes the weight of the sensitivity of the (i_1,j_1) th design element contributed to the modified sensitivity $\frac{\partial \tilde{c}}{\partial \bar{\rho}_{i,j}}$ of the (i,j) th design element, which is computed by

$$\frac{\partial \tilde{c}}{\partial \bar{\rho}_{i,j}} = \frac{1}{\max(\gamma, \bar{\rho}_{i,j}) \sum_{(i_2,j_2) \in \mathbb{N}_{(i,j)}} w_{(i,j),(i_2,j_2)}} \cdot \sum_{(i_1,j_1) \in \mathbb{N}_{(i,j)}} \left[w_{(i,j),(i_1,j_1)} \cdot \bar{\rho}_{i_1,j_1} \cdot \frac{\partial c}{\partial \bar{\rho}_{i_1,j_1}} \right], \quad (21)$$

with γ ($\gamma = 10^{-3}$) used to avoid division by zero, $\frac{\partial c}{\partial \bar{\rho}_{i,j}}$ derived from Eqs. (17) and (18) and is formulated as

$$\frac{\partial c}{\partial \bar{\rho}_{i,j}} = -penal \cdot (\bar{\rho}_{i,j})^{penal-1} \cdot (E_0 - E_{\min}) \cdot (\mathbf{u}_{i,j})^T \cdot \mathbf{k}_0 \cdot \mathbf{u}_{i,j}, \quad (22)$$

Moreover, for an arbitrary design variable $\rho_{i,j}$, the sensitivity of the volume constraint function is written in

$$\frac{\partial V}{\partial \bar{\rho}_{i,j}} = 1. \quad (23)$$

3.3 Sensitivity analysis by the splitting of explicit filter

The splitting process for the tensor product-based explicit filter is basically similar to the splitting process for the implicit filter, except that the sum of the weights over the influence region must be computed and stored for each design variable. According to the tensor product structure shown in Eq. (19) and the local support property of $w_{i,i_1}(x)$ shown in Eq. (20), Eq. (21) can be rewritten into

$$\frac{\partial \tilde{c}}{\partial \bar{\rho}_{i,j}} = \frac{1}{\max(\gamma, \bar{\rho}_{i,j}) H_{i,j}} \cdot \sum_{i_1=\max(1,i-\lceil r_{\min} \rceil)}^{\min(ne_x,i+\lceil r_{\min} \rceil)} \sum_{j_1=\max(1,j-\lceil r_{\min} \rceil)}^{\min(ne_y,j+\lceil r_{\min} \rceil)} \left[w_{i,i_1}(x) \cdot w_{j,j_1}(y) \cdot \bar{\rho}_{i_1,j_1} \cdot \frac{\partial c}{\partial \bar{\rho}_{i_1,j_1}} \right], \quad (24)$$

with

$$H_{i,j} = \sum_{(i_2,j_2) \in \mathbb{N}_{(i,j)}} w_{(i,j),(i_2,j_2)} = \sum_{i_2=\max(1,i-\lceil r_{\min} \rceil)}^{\min(ne_x,i+\lceil r_{\min} \rceil)} \sum_{j_2=\max(1,j-\lceil r_{\min} \rceil)}^{\min(ne_y,j+\lceil r_{\min} \rceil)} w_{i,i_2}(x) \cdot w_{j,j_2}(y), \quad (25)$$

where $H_{i,j}$ is the sum of the weights over the influence region of the (i,j) th design element. Naturally, identical to the matrix form presented in Eq. (12) for the implicit filter, the sensitivities of the objective function defined in Eq. (24) for the explicit filter can be reformulated as

$$\begin{bmatrix} \frac{\partial \tilde{c}}{\partial \bar{\rho}_{1,1}} & \frac{\partial \tilde{c}}{\partial \bar{\rho}_{1,2}} & \cdots & \frac{\partial \tilde{c}}{\partial \bar{\rho}_{1,ne_y}} \\ \frac{\partial \tilde{c}}{\partial \bar{\rho}_{2,1}} & \frac{\partial \tilde{c}}{\partial \bar{\rho}_{2,2}} & \cdots & \frac{\partial \tilde{c}}{\partial \bar{\rho}_{2,ne_y}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \tilde{c}}{\partial \bar{\rho}_{ne_x,1}} & \frac{\partial \tilde{c}}{\partial \bar{\rho}_{ne_x,2}} & \cdots & \frac{\partial \tilde{c}}{\partial \bar{\rho}_{ne_x,ne_y}} \end{bmatrix} = \mathbf{H}_{ne_x \times ne_x}(x) \cdot \mathbf{dc}_{ne_x \times ne_y} \cdot \mathbf{H}_{ne_y \times ne_y}(y), \quad (26)$$

with

$$\begin{aligned}
& d\mathbf{c}_{n_e \times n_{e_y}} \\
& = \begin{bmatrix} \frac{1}{\max(\gamma, \bar{\rho}_{1,1}) H_{1,1}} \cdot \bar{\rho}_{1,1} \cdot \frac{\partial c}{\partial \bar{\rho}_{1,1}} & \frac{1}{\max(\gamma, \bar{\rho}_{1,2}) H_{1,2}} \cdot \bar{\rho}_{1,2} \cdot \frac{\partial c}{\partial \bar{\rho}_{1,2}} & \cdots & \frac{1}{\max(\gamma, \bar{\rho}_{1,n_{e_y}}) H_{1,n_{e_y}}} \cdot \bar{\rho}_{1,n_{e_y}} \cdot \frac{\partial c}{\partial \bar{\rho}_{1,n_{e_y}}} \\ \frac{1}{\max(\gamma, \bar{\rho}_{2,1}) H_{2,1}} \cdot \bar{\rho}_{2,1} \cdot \frac{\partial c}{\partial \bar{\rho}_{2,1}} & \frac{1}{\max(\gamma, \bar{\rho}_{2,2}) H_{2,2}} \cdot \bar{\rho}_{2,2} \cdot \frac{\partial c}{\partial \bar{\rho}_{2,2}} & \cdots & \frac{1}{\max(\gamma, \bar{\rho}_{2,n_{e_y}}) H_{2,n_{e_y}}} \cdot \bar{\rho}_{2,n_{e_y}} \cdot \frac{\partial c}{\partial \bar{\rho}_{2,n_{e_y}}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\max(\gamma, \bar{\rho}_{n_{e_x},1}) H_{n_{e_x},1}} \cdot \bar{\rho}_{n_{e_x},1} \cdot \frac{\partial c}{\partial \bar{\rho}_{n_{e_x},1}} & \frac{1}{\max(\gamma, \bar{\rho}_{n_{e_x},2}) H_{n_{e_x},2}} \cdot \bar{\rho}_{n_{e_x},2} \cdot \frac{\partial c}{\partial \bar{\rho}_{n_{e_x},2}} & \cdots & \frac{1}{\max(\gamma, \bar{\rho}_{n_{e_x},n_{e_y}}) H_{n_{e_x},n_{e_y}}} \cdot \bar{\rho}_{n_{e_x},n_{e_y}} \cdot \frac{\partial c}{\partial \bar{\rho}_{n_{e_x},n_{e_y}}} \end{bmatrix}_{n_e \times n_{e_y}}, \quad (27)
\end{aligned}$$

and

$$\mathbf{H}_{n_e \times n_{e_x}}(i, :) = [0, \dots, w_{i, i-\lceil r_{\min} \rceil}(x), \dots, w_{i, i+\lceil r_{\min} \rceil}(x), 0, \dots], \quad \lceil r_{\min} \rceil < i \leq n_{e_x} - \lceil r_{\min} \rceil, \quad (28)$$

$$\mathbf{H}_{n_e \times n_{e_y}}(j, :) = [0, \dots, w_{j, j-\lceil r_{\min} \rceil}(y), \dots, w_{j, j+\lceil r_{\min} \rceil}(y), 0, \dots], \quad \lceil r_{\min} \rceil < j \leq n_{e_y} - \lceil r_{\min} \rceil, \quad (29)$$

where $\mathbf{H}_{n_e \times n_{e_x}}$ and $\mathbf{H}_{n_e \times n_{e_y}}$ are the weight matrices along with x and y directions, respectively. Similar to Eq. (16), the matrix form of Eq. (25) is formulated as

$$\begin{bmatrix} H_{1,1} & H_{1,2} & \cdots & H_{1,n_{e_y}} \\ H_{2,1} & H_{2,2} & \cdots & H_{2,n_{e_y}} \\ \vdots & \vdots & \ddots & \vdots \\ H_{n_{e_x},1} & H_{n_{e_x},2} & \cdots & H_{n_{e_x},n_{e_y}} \end{bmatrix} = \begin{bmatrix} \sum_{i_1=1}^{1+\lceil r_{\min} \rceil} w_{1,i_1}(x) \\ \vdots \\ \sum_{i_1=i-\lceil r_{\min} \rceil}^{i+\lceil r_{\min} \rceil} w_{i,i_1}(x) \\ \vdots \\ \sum_{i_1=n_{e_x}-\lceil r_{\min} \rceil}^{n_{e_x}} w_{n_{e_x},i_1}(x) \end{bmatrix} \cdot \begin{bmatrix} \sum_{j_1=1}^{1+\lceil r_{\min} \rceil} w_{1,j_1}(y) & \cdots & \sum_{j_1=j-\lceil r_{\min} \rceil}^{j+\lceil r_{\min} \rceil} w_{j,j_1}(y) & \cdots & \sum_{j_1=n_{e_y}-\lceil r_{\min} \rceil}^{n_{e_y}} w_{n_{e_y},j_1}(y) \end{bmatrix}. \quad (30)$$

3.4 Computing and storing of the weight matrices of explicit filter

For a traditional explicit sensitivity filter, the computing time consists of the calculating time of centroid distances between adjacent elements and the assembling time of the sparse weight matrix. The computing complexities of the traditional explicit sensitivity filter for 2D and 3D TO problems are $O(n^2 \lceil r_{\min} \rceil^2)$ and $O(n^3 \lceil r_{\min} \rceil^3)$, respectively, provided that the number of design elements along all axial directions takes the same constant value n . Besides, the storing space complexities of the traditional explicit sensitivity filter for 2D and 3D TO problems are $O(n^2 \lceil r_{\min} \rceil^2)$ and $O(n^3 \lceil r_{\min} \rceil^3)$, respectively. As for the tensor product-based explicit sensitivity filter, the computing time consists of the calculating time of

centroid distances between adjacent elements along all axial directions and the assembling time of the sparse weight matrices of all axial directions. Similar to the traditional explicit sensitivity filter case, the computing complexities of the traditional explicit sensitivity filter are $O(n \lceil r_{\min} \rceil)$, and the storing space complexities of the tensor product structure-based explicit sensitivity filter for 2D and 3D TO problems are $O(n \lceil r_{\min} \rceil)$. Besides, the comparisons in computing time and storage burden between the traditional explicit sensitivity filter and the proposed tensor product structure-based ones are illustrated in Figs. 8 and 9 for 3D TO problems, respectively.

4 Numerical examples with IGA-based TO method

To examine the effectiveness of the proposed splitting technique for implicit B-splines filter, this section presents two canonical TO problems solved by IGA-based TO method, with the geometric and physical parameters chosen as the dimensionless quantities. The Young's modulus for the solid and void material phases are 1 and 10^{-9} , respectively; and the Poisson's ratio is specified to 0.3. If without additional specification, the convergence criterion of the presented numerical examples in this work is prescribed to Iteration ≤ 301 or $\theta_{\text{Iterative_step}} \leq 0.01$, with Iteration and $\theta_{\text{Iterative_step}}$ denoting the current iterative step and the maximum derivation of design variables on consecutive iterative steps, respectively. In addition, the software environment is MATLAB R2021a, and the hardware test environment is a workstation with Xeon(R) Gold 5120 CPU @ 2.20 GHz 2.19 GHz Intel core and 64 GB RAM for the benchmarks presented in this paper.

4.1 Quarter annulus cantilever

To verify the effectiveness of the proposed splitting technique for the NURBS implicit filter, the quarter annulus cantilever is treated as the first benchmark in this work, of which the problem setting is illustrated in

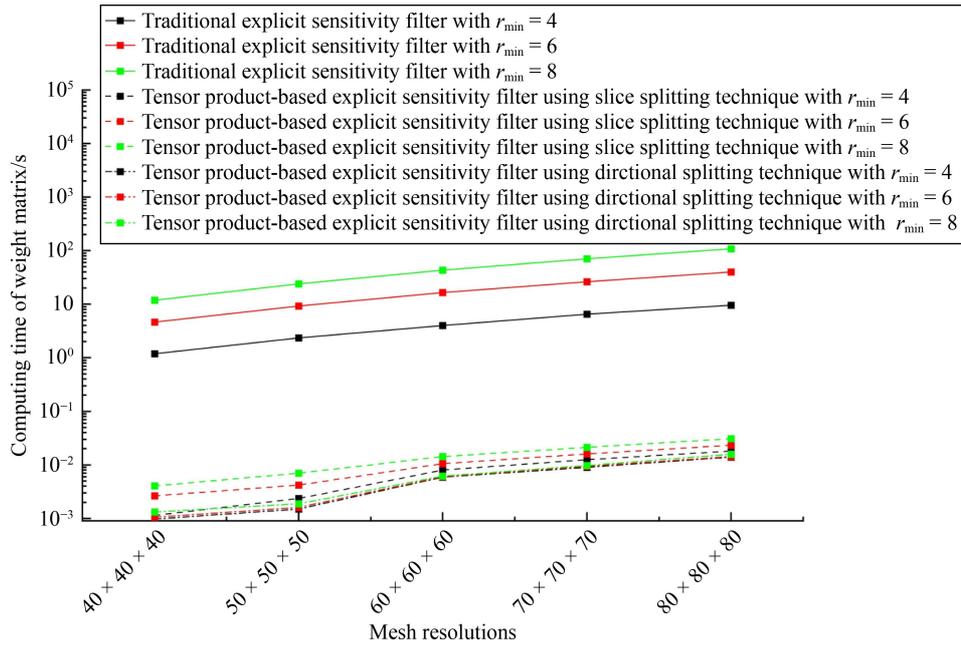


Fig. 8 Illustration of the comparisons in computing time between the traditional sensitivity filter and the proposed splitting ones.

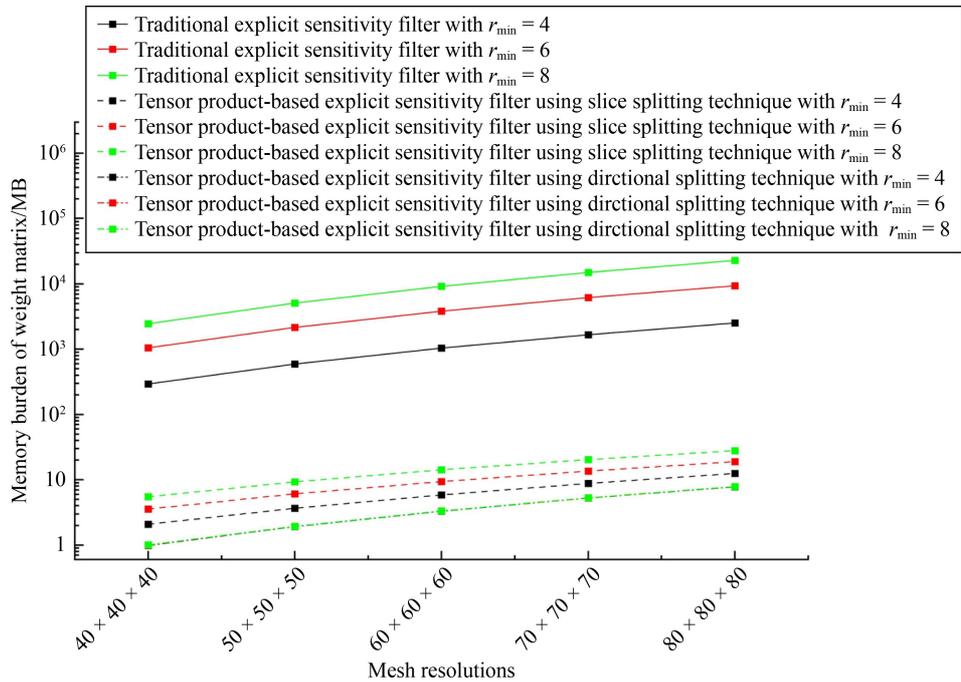


Fig. 9 Illustration of the comparison in storage burden between the traditional sensitivity filter and the proposed splitting ones.

Fig. 10. The external load of the quarter annulus cantilever is applied to the left-top corner, the right-bottom edge is fixed as the displacement condition, and the upper limit is set to 0.4 for the volume fraction of the solid material. Besides, the prescribed ratio of the inner circle radius to the outer circle radius is 0.5. A NURBS mesh with the size of 800 × 400 is used to discretize the design domain to obtain the structural compliance, where the bi-quadratic plane-stress elements are used. To parametrize the density design field of the design domain,

the densities of control points are treated as the nodal design variables of TO, and the size of the NURBS design mesh is 200 × 400, of which three different degree combinations, namely, 10 × 10, 20 × 20, and 30 × 30 are considered to compare the differences between traditional NURBS implicit filter and the decomposed NURBS implicit filter by using the proposed splitting technique.

The results optimized by IGA-based TOs using six different NURBS implicit filters mentioned above are presented in Fig. 11, from which the decomposed

NURBS implicit filter established on the splitting technique of tensor product structure can generate identical material layouts as they are produced by the traditional one, which is invariant to the degree combination of NURBS. Figure 12 illustrates the convergence histories for the traditional and decomposed NURBS implicit filter under three different degree combinations of NURBS. According to the curves presented in Fig. 12, the decomposed NURBS implicit filter shares the same convergence history as the traditional NURBS implicit filter, independent of the degree combination of NURBS of the isogeometric design mesh. Based on these two previous observations,

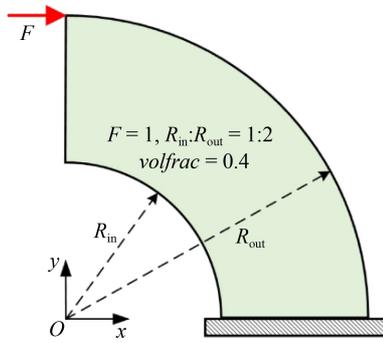


Fig. 10 Illustration of the problem setting for the quarter annulus cantilever design problem.

the decomposed NURBS implicit filter is equivalent to the traditional ones, and the validity of the sensitivity analysis reformulated by the decomposed implicit filter is verified for the IGA-based TO method.

The updating time of design variables for each iterative step between the traditional and the decomposed NURBS implicit filters with three different degree combinations is compared in Fig. 13. The efficiency of sensitivity analysis shows that the decomposed NURBS implicit filter outperforms the traditional one, and the improved efficiency is enhanced with the degrees of NURBS of isogeometric design mesh. The underlying reason for the improved sensitivity analysis is that the computational cost related to the modified sensitivities of all design variables is reduced because of the original weight matrix in a large scale decomposed into two submatrices in a small scale. Figure 14 depicts the comparison results in memory burden and pre-computing time between the traditional and the decomposed NURBS implicit filter for the weight matrix of all isogeometric design elements. As shown in Fig. 14, the memory burden is reduced by a factor that ranges from 179 to 1245.3. Moreover, the pre-computed time is decreased by a factor that belongs to the interval of [14.1, 28.3] when the traditional NURBS implicit filter is replaced by the decomposed one by means of the proposed splitting technique of the tensor product structure. In a word, the proposed decomposed NURBS implicit filter can be used to replace the

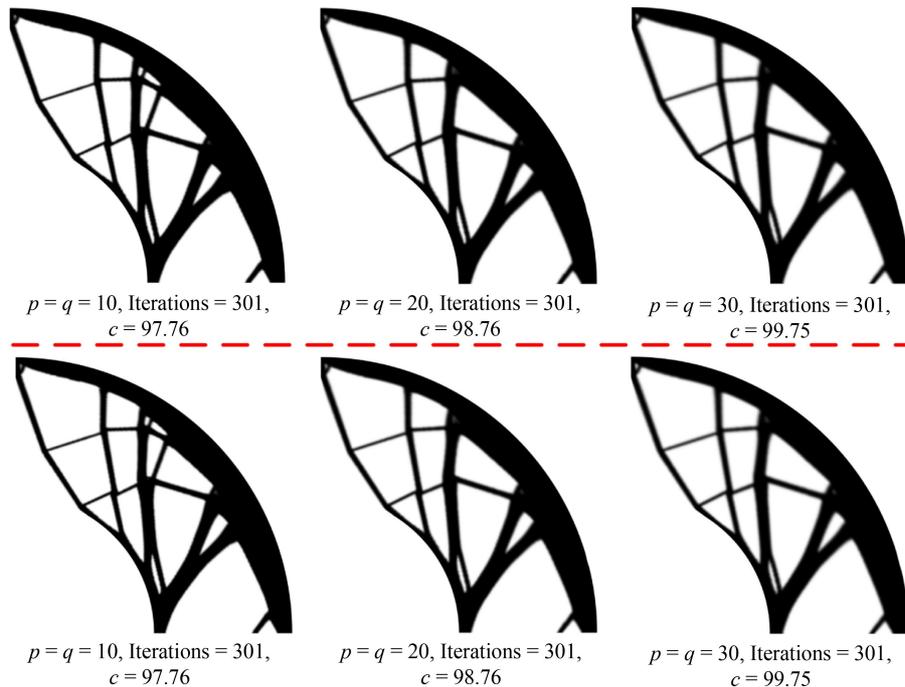


Fig. 11 Illustration of the optimized structures for the quarter annulus cantilever design problem: the results obtained by isogeometric analysis-based topology optimization method with traditional non-uniform rational B-splines filter under three different degree combinations (on the top row); the results obtained by isogeometric analysis-based topology optimization method with the proposed decomposed non-uniform rational B-splines filter by the splitting technique under three different degree combinations (on the bottom row).

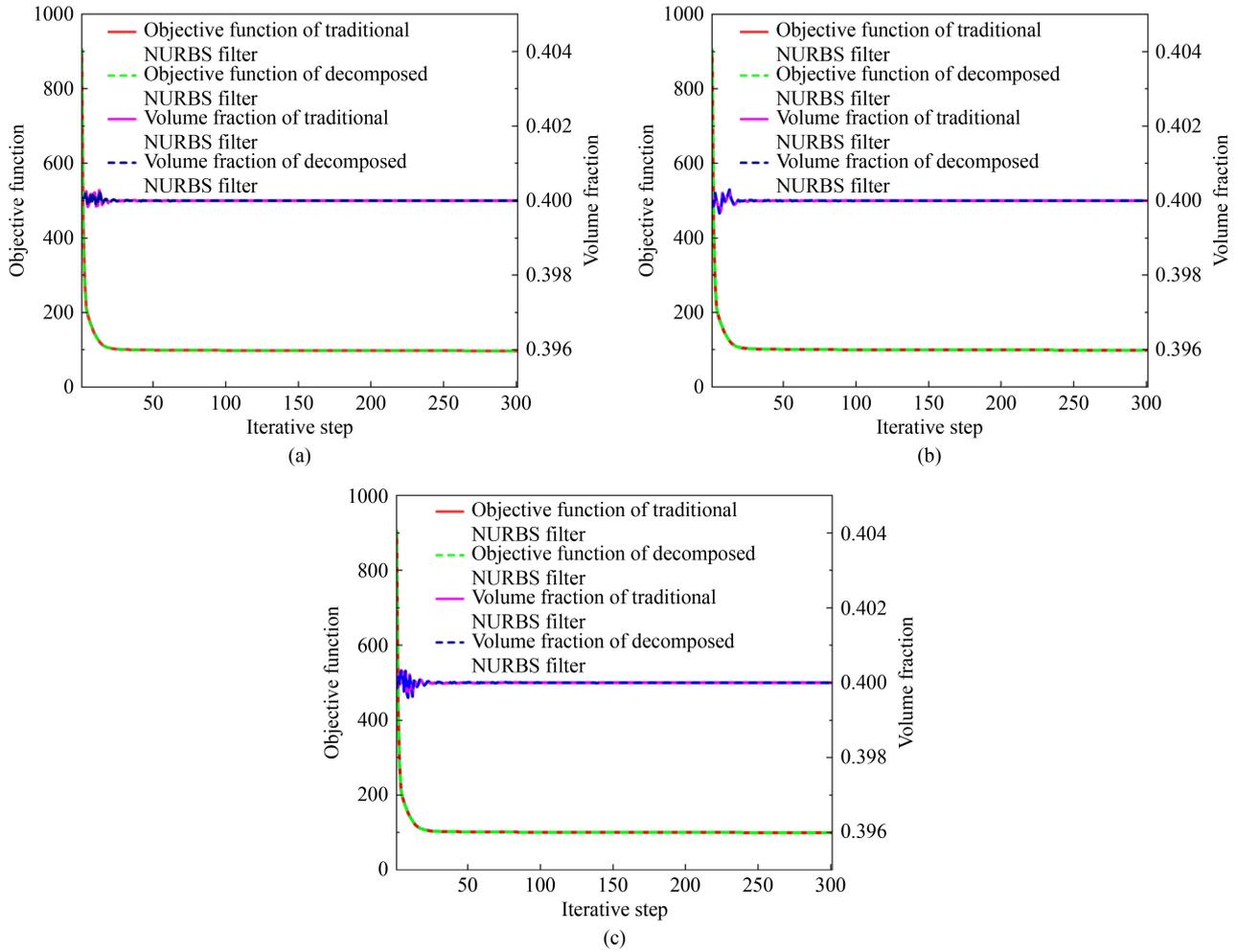


Fig. 12 Comparisons in convergence histories of isogeometric analysis-based topology optimization method between the traditional and the proposed decomposed NURBS implicit filters under three different degree combinations: (a) 10×10 -degree combination, (b) 20×20 -degree combination, and (c) 30×30 -degree combination. NURBS: non-uniform rational B-spline.

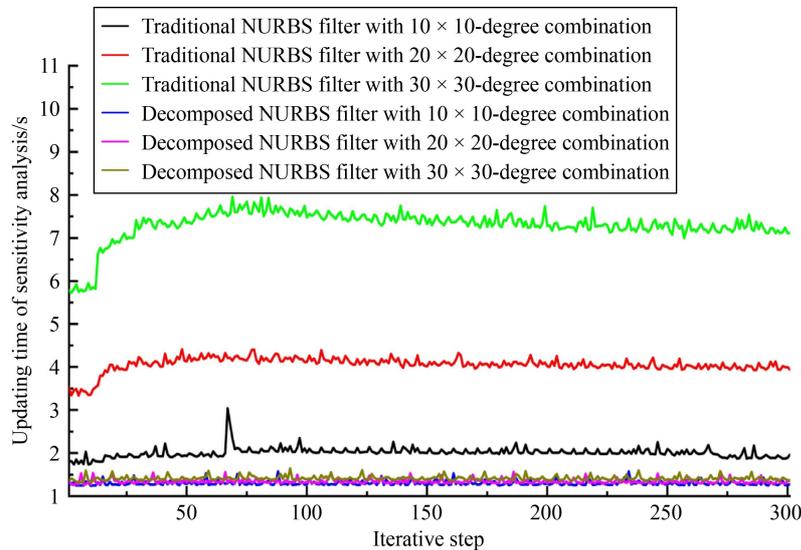


Fig. 13 Comparisons in updating time for each iterative step of isogeometric analysis-based topology optimization methods between the traditional and the proposed decomposed NURBS implicit filters under three different degree combinations. NURBS: non-uniform rational B-spline.

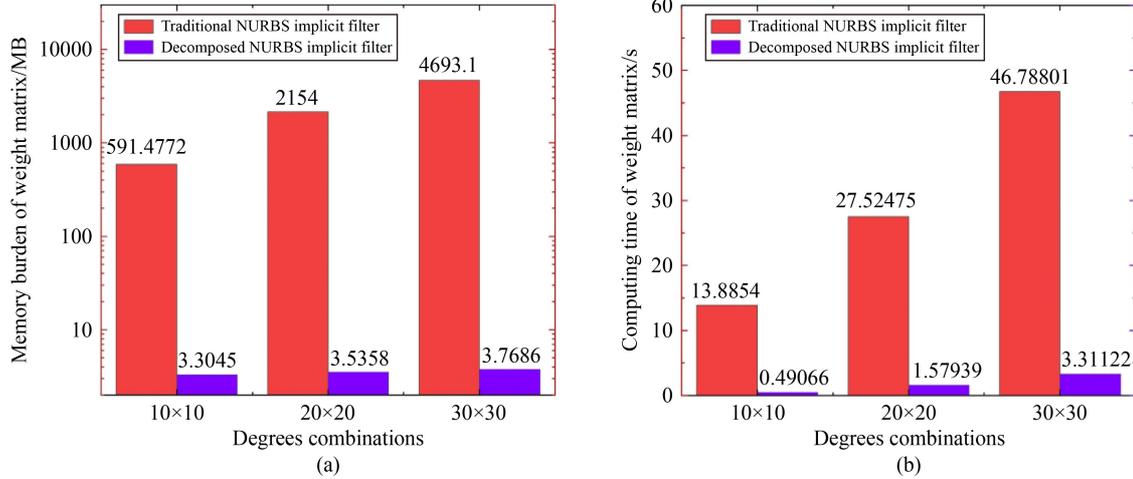


Fig. 14 Comparisons in storage and computational efficiencies between the traditional and the proposed decomposed NURBS implicit filters under three different degree combinations: (a) comparisons in memory burden and (b) comparisons in pre-computing time of weight matrix. NURBS: non-uniform rational B-spline.

traditional one for the IGA-based TO method without altering the optimized results and convergence history while reducing the updating time of design variables and memory burden, as well as the computing time of the pre-computed weight matrix significantly. Finally, the efficiency of NURBS implicit filter is improved massively for 2D curved design problems solved by IGA-based TO method, which resulted from the splitting of the NURBS tensor product structure.

4.2 Three-dimensional cantilever

A 3D cantilever illustrated in Fig. 15 is used to validate the effectiveness of the proposed splitting technique in 3D TO problems, which are optimized by IGA-based TO method with B-spline implicit filters. As presented in Fig. 15, the left face is fixed, and a uniformly distributed load is applied to the right-bottom edge as the boundary condition for the 3D cantilever. The prescribed volume fraction of the solid material is specified to 0.3. Moreover, the IGA mesh using tri-quadratic B-spline elements is treated as the analysis mesh, of which the size is set to $60 \times 30 \times 30$. The mesh size of the B-spline design meshes is prescribed to $30 \times 15 \times 15$, which is successively refined by the h-refinement technique of B-splines, while the degrees of B-splines are set to $20 \times 20 \times 20$.

Figure 16 illustrates the optimized structures using three different B-spline implicit filters described in Subsection 2.2. The optimized structures and the objective functions are exactly identical for 3D cantilever design problem using these three different forms of B-spline implicit filters. Besides, the convergence history curves of the objective function and volume fraction are depicted in Fig. 17 for 3D cantilever optimized by IGA-based TO methods under different filters, indicating that

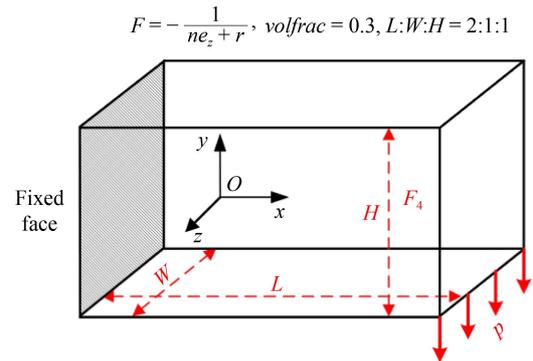


Fig. 15 Illustration of the problem setting for 3D cantilever optimized by multi-resolution isogeometric analysis-based topology optimization method with B-spline implicit filters.

the B-spline implicit filters that use slice and directional splitting techniques shares the same convergence process with the traditional ones for IGA-based TO method. Therefore, similar to the conclusion drawn in the previous benchmark, the decomposed B-spline implicit filters by the slice and directional splitting techniques are equivalent to the traditional one for 3D design problems that are optimized by IGA-based TO method.

Moreover, Fig. 18 presents the comparisons in the updating time of sensitivity analysis at each iterative step between the traditional B-spline implicit filter and the decomposed ones generated by slice and directional splitting techniques. From the viewpoint of the updating efficiency of the nodal design variables, the decomposed B-spline implicit filters outperform the traditional one. According to the comparison results illustrated in Fig. 19, the memory burden and the pre-computing time involved in the weight matrix of the B-spline implicit filter are reduced significantly when the traditional implicit filter is replaced by the decomposed ones using two different

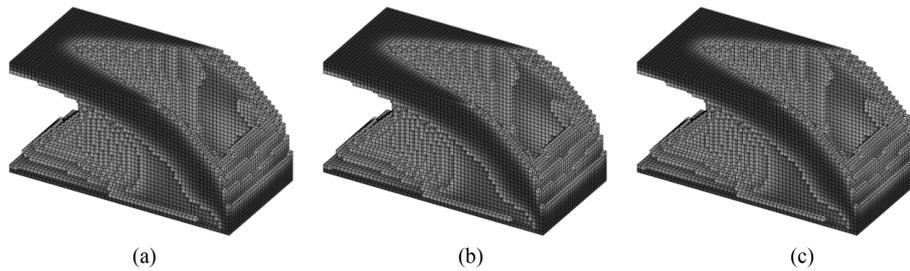


Fig. 16 Illustrations of the optimized structures for the 3D cantilever beam: (a) result obtained by traditional B-spline implicit filter, (b) result obtained by decomposed B-spline implicit filter using slice splitting technique, and (c) result obtained by decomposed B-spline implicit filter using directional splitting technique. Iterations = 301, $c = 4.8034$.

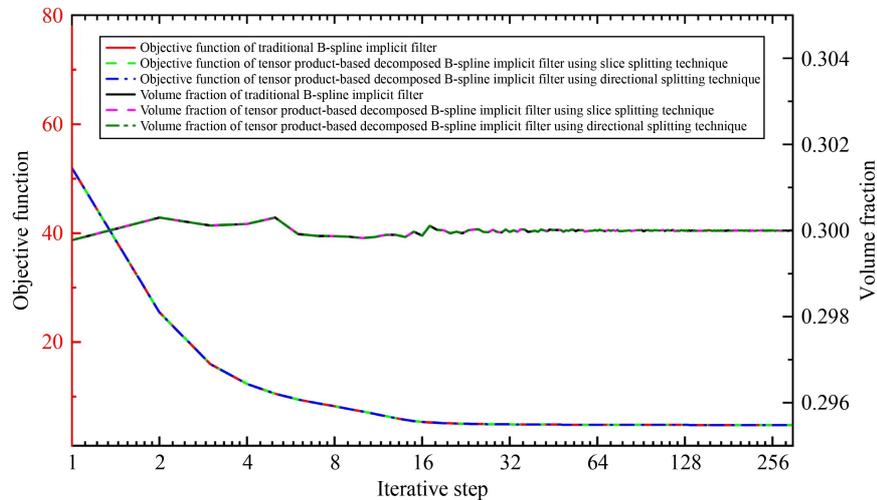


Fig. 17 Comparisons in the objective function and volume fraction of traditional B-spline implicit filters, and the tensor product-based decomposed B-spline implicit filters using slice and directional splitting technique.

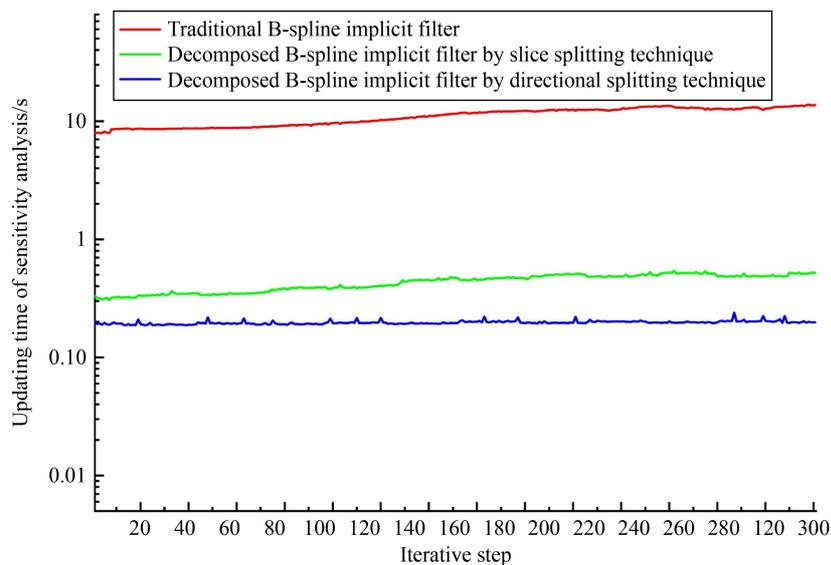


Fig. 18 Comparisons in updating time of sensitivity analysis for each step of the traditional and the proposed decomposed B-spline implicit filters by the slice and directional splitting techniques.

splitting techniques. Moreover, compared with the traditional B-spline filter, the memory burden and pre-computing time of the decomposed filter are reduced by 99.841% and 99.68% by applying the slice splitting technique and the associated reductions are up to 99.999% for the decomposed filter using the directional

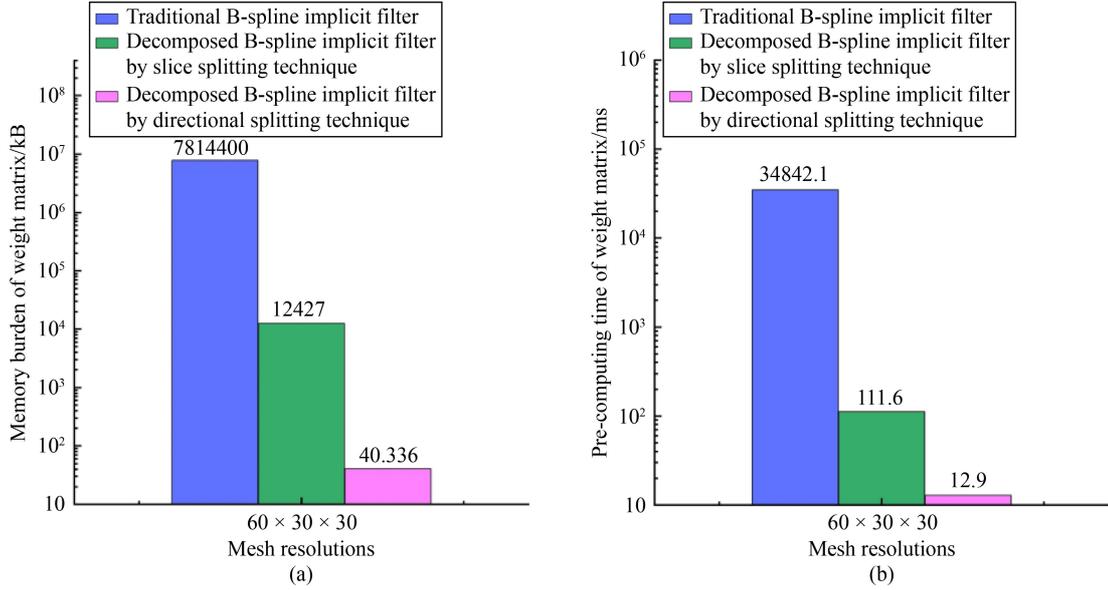


Fig. 19 Comparisons between the traditional and the decomposed B-spline implicit filters by slice and directional splitting technique for (a) the memory burden of weight matrix and (b) the pre-computing time of weight matrix.

splitting technique. Meanwhile, the directional splitting technique is a more appropriate choice than the slice one for the decomposing of B-spline tensor product structure because of the improved efficiency in updating design variables and the memory burden and the pre-computing of the weight matrix. In a word, the traditional B-spline implicit filter can be equivalently replaced by the decomposed ones using the splitting techniques proposed in this work with identical optimized designs and convergence history while decreasing the memory burden and computing time of the weight matrix, as well as the updating time for the nodal design variables significantly. Therefore, the efficiency of B-spline implicit filter for 3D design problems is massively improved by our proposed splitting techniques for the tensor product structure of B-splines.

5 Numerical examples with FEM-based TO method

To validate the effectiveness of the proposed explicit sensitivity filter in the tensor product form and the splitting technique for the explicit filter, this section presents two canonical TO problems solved by FEM-based TO method. The material properties, convergence criterion, and the execution environment are identical to the parameters presented in Section 4 for TO problems optimized by IGA-based TO method.

5.1 Short beam problem

As a well-known benchmark in TO, the short beam loaded on the middle of right edge is presented in this

work to verify the effectiveness of our proposed explicit sensitivity filter in Section 3 along with the splitting techniques of the tensor product structure. Figure 20 depicts the problem setting for the short beam, which is clamped at the left edge and subjected to a vertically downward external load with a magnitude equal to 1. The aspect ratio is 2, and the volume percentage of the solid material is set to 0.4 for the design domain. Bi-linear plane-stress FEM elements are used to discretize the design domain.

Subsequently, Fig. 21 presents the optimized results obtained by the traditional and the decomposed explicit sensitivity filters under three different mesh resolutions, namely, 800×400 , 1200×600 , and 2000×1000 . The optimized design of the left bottom figure in Fig. 21 is lacking for traditional explicit sensitivity filter under 2000×1000 mesh resolutions because the memory burden of the FEM solver is prohibitive for our workstation. According to the results depicted in Fig. 21, the optimized results generated by the two different explicit sensitivity filters are basically identical, even

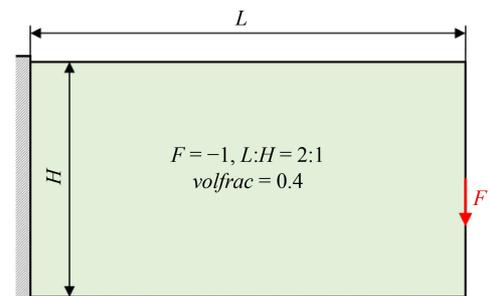


Fig. 20 Illustration of the problem configurations of short beam.

though slight differences exist in the filter radius. In addition, the comparisons in the convergence history between the traditional filter and the decomposed one are shown in Fig. 22, from which we can observe that the decomposed explicit filter shares basically identical

convergent process with the traditional explicit filter, independent of the mesh resolution.

The comparison in the updating time between the two different explicit filters is illustrated in Fig. 23 for the sensitivity analysis of per iterative step, which reveals

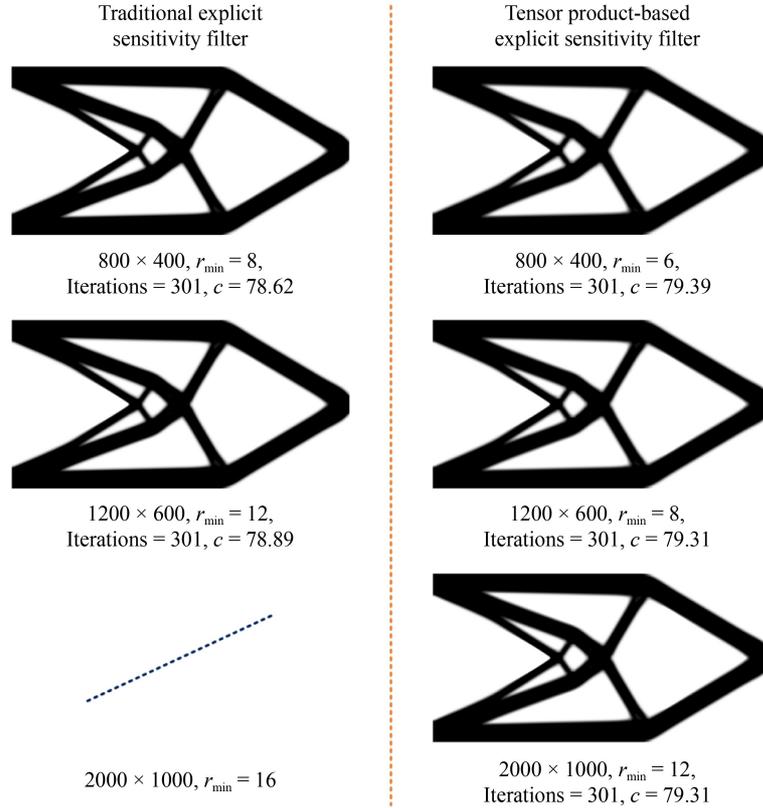


Fig. 21 Illustration of the optimized structures for the short beam problem by finite element method-based topology optimization methods using different filters: traditional explicit sensitivity filter (on the left column); decomposed explicit sensitivity filter (on the right column).

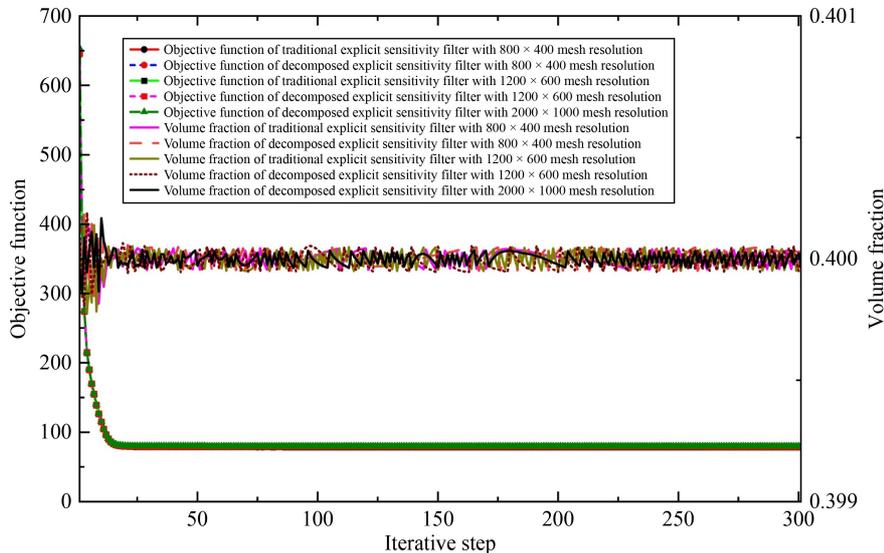


Fig. 22 Comparisons in the convergence curves of the finite element method-based topology optimization method between the traditional and the decomposed explicit sensitivity filter under different mesh resolutions.

that decomposed explicit sensitivity filter using the proposed splitting technique is superior to the traditional one, and the enhanced efficiency performance is improved with the increase in the mesh resolution. Subsequently, the comparisons in memory burden and pre-computing time of the weight-matrix for the two different explicit filters are illustrated in Fig. 24. As shown in Fig. 24, the memory burden of the decomposed explicit filter is reduced by a factor that varies from 216.9 to 759.5, and the pre-computing time of the weight matrix is decreased by a ratio that ranges from 72.4 and 213.4 compared with the traditional explicit sensitivity filter. Based on the observations presented above, either the updating efficiency of the design variables or the memory

and computational efficiency of the weight matrix are largely improved by replacing the traditional explicit filter with the decomposed one obtained by the splitting technique simultaneously. Hence, the splitting technique of tensor product structure is an effective way of enhancing the efficiency of the explicit filter for 2D design problems.

5.2 3D cantilever

To examine the superiorities of the proposed decomposed explicit sensitivity filter by the splitting technique over the traditional explicit sensitivity filter in 3D TO problems, 3D cantilever subjected to external torsional

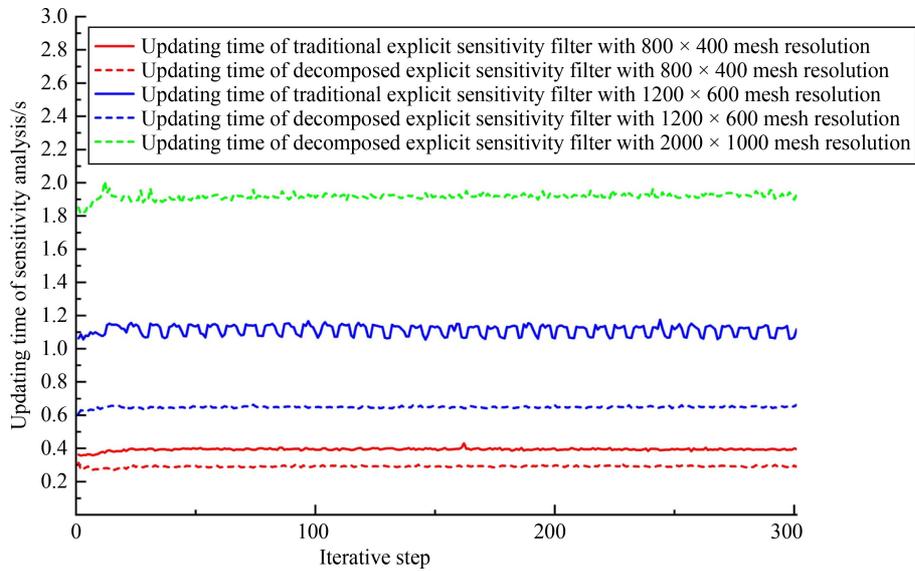


Fig. 23 Comparisons in the updating time of design variables for each step of finite element method-based topology optimization method between the traditional and the proposed decomposed explicit sensitivity filter under three different mesh resolutions.

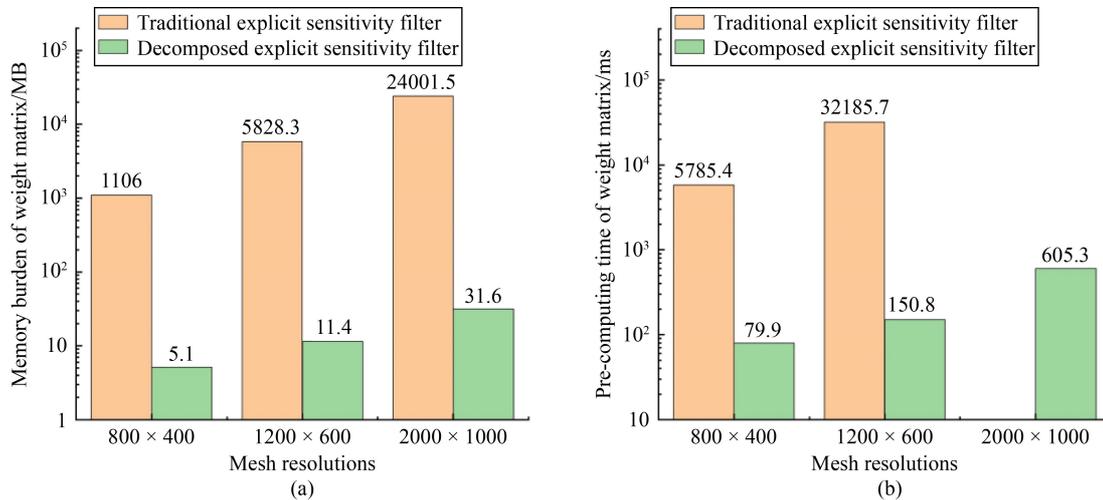


Fig. 24 Comparisons between the traditional and the decomposed explicit sensitivity filter for (a) the memory burden of weight matrix and (b) the pre-computing time of weight matrix.

forces is presented in this subsection, of which the basic problem configurations are illustrated in Fig. 25 and solved by the multi-resolution TO (MTO) method. As presented in Fig. 25, the left face is clamped, and the four external torsional forces are applied on the four corners of the right face and located on the right face, as well as the upper limit in volume fraction of solid material is prescribed to 0.15. The mesh size of MTO design and analysis meshes are initialized to $30 \times 15 \times 15$, which are

$$|F_1| = |F_2| = |F_3| = |F_4| = \sqrt{2}, \theta_1 = \theta_2 = \theta_3 = \theta_4 = \frac{\pi}{4}$$

$$\text{volfrac} = 0.15, L:W:H = 2:1:1$$

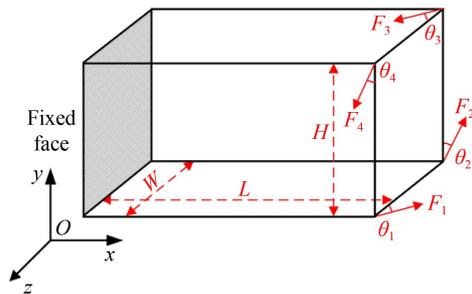


Fig. 25 Illustration of the problem setting for 3D cantilever subjected to multiple external loads.

globally refined during the optimization process. All the elements are tri-linear FEM elements.

Two intermediate designs at the iterative steps with the triggered global refinement of the design mesh, and the final design are depicted in Fig. 26 for the 3D cantilever optimized by three different explicit sensitivity filters. As the converged designs shown in Fig. 26, the optimized results are identical for the explicit sensitivity filters decomposed by the slice and directional splitting techniques of tensor product structure, which are basically identical to the ones obtained from the traditional explicit sensitivity filter. Moreover, the convergence curves shown in Fig. 27 coincide for the decomposed explicit filters, which are similar to the associated curves of the traditional explicit filter. In a word, explicit sensitivity filters by the slice and directional splitting techniques can be applied to 3D TO problems equivalently.

Afterwards, the sensitivity analysis updating time for each iterative step is compared in Fig. 28 between the traditional and the decomposed explicit filters. As the design mesh refined, the updating time of the decomposed filters becomes less than that of the traditional filter for the same design mesh because of the

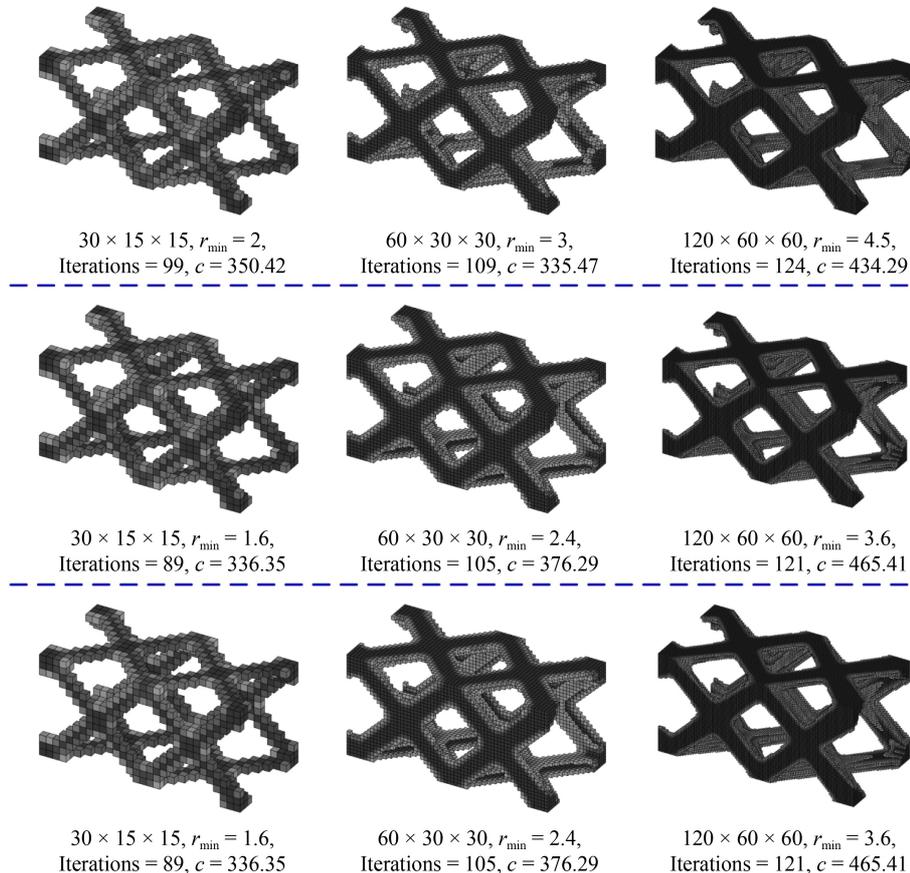


Fig. 26 Illustration of some intermediate and the final designs of three examples for 3D cantilever beam: traditional explicit sensitivity filter (on the top row); decomposed explicit sensitivity filter using the slice splitting technique of tensor product structure (on the middle row); and decomposed explicit sensitivity filter using the directional splitting technique of tensor product structure (on the bottom row).

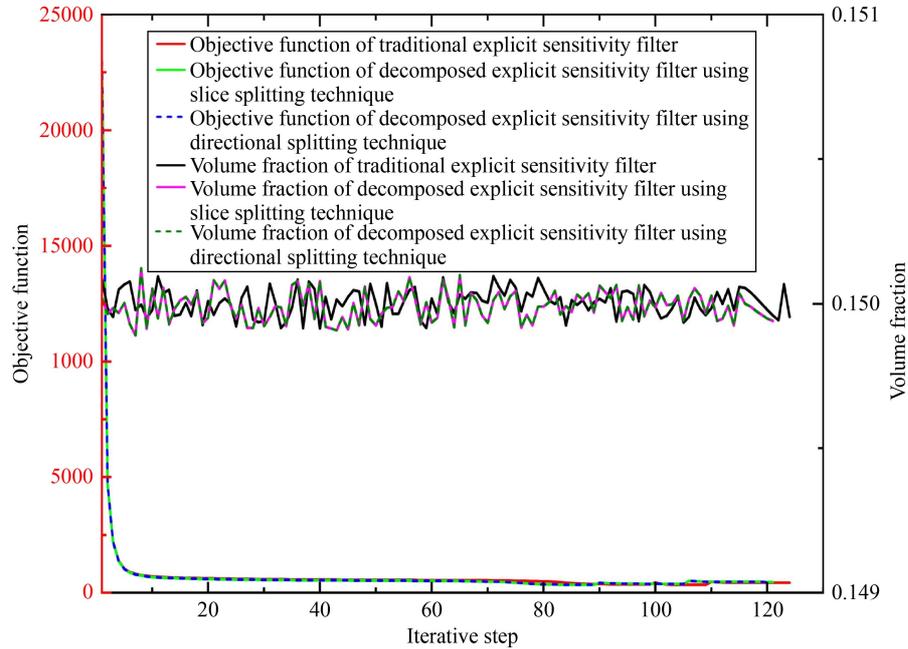


Fig. 27 Comparisons in objective function and volume fraction of traditional filter and the decomposed explicit sensitivity filters using the slice and directional splitting techniques.

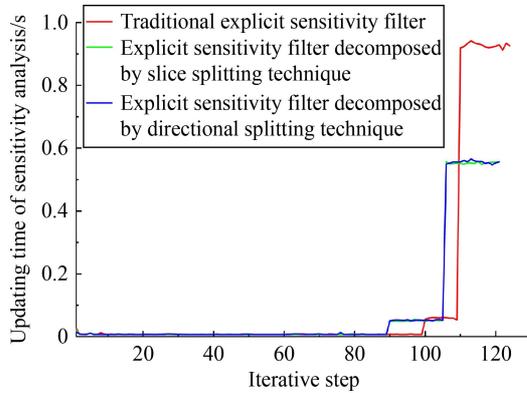


Fig. 28 Comparisons in updating time of sensitivity for each step among three different explicit filters for multi-resolution topology optimization.

reducing scale of the weight matrix. Moreover, Fig. 29 illustrates the memory burden and computing time of the weight matrix for the MTOPs using three different explicit sensitivity filters. Compared with the traditional explicit sensitivity filter, the explicit sensitivity filter decomposed by slice splitting technique reduces the memory burden by 94.38%, 98.57%, and 99.75% and pre-computing time by 79.38%, 98.01%, and 99.93% for the three different design mesh resolutions of $30 \times 15 \times 15$, $60 \times 30 \times 30$, and $120 \times 60 \times 60$, respectively. The corresponding improvements for the memory burden of the weight matrix are 96.37%, 99.2%, and 99.86%. And 56.7%, 97.89%, and 99.95% for the pre-computing efficiency of the weight matrix, when the MTOP utilizes the explicit filter decomposed by the directional splitting

technique. Thus, the efficiency of the explicit sensitivity filter can be enhanced significantly by the proposed tensor product-based explicit filter decomposed by the splitting techniques of the tensor product structure. Therefore, the proposed tensor product-based explicit filter is a much more improved filtering technique than the traditional one for 3D design problems solved by FEM-based MTOP method in combination with the proposed splitting techniques for the tensor product structure.

6 Conclusions

We have developed the splitting technique for B-splines/NURBS implicit filters for TO problems in the framework of IGA, where the traditional NURBS/B-spline weight matrix is decomposed into several smaller separated weight submatrices along with the parametric directions. Subsequently, the sensitivity analysis of the IGA-based TO model is reformulated in terms of the factorized submatrices, which is equivalent to that derived from the original weight matrix induced from the tensor product structure of B-splines/NURBS. Then, we devise an explicit sensitivity filter in the tensor product form for FEM-based TO method, where the weight between two adjacent elements is calculated by the tensor product of their centroid axial distances. Similar to the implicit filters, the sensitivity analysis of the proposed tensor product explicit filter is established on the basis of the proposed splitting techniques for FEM-based TO method.

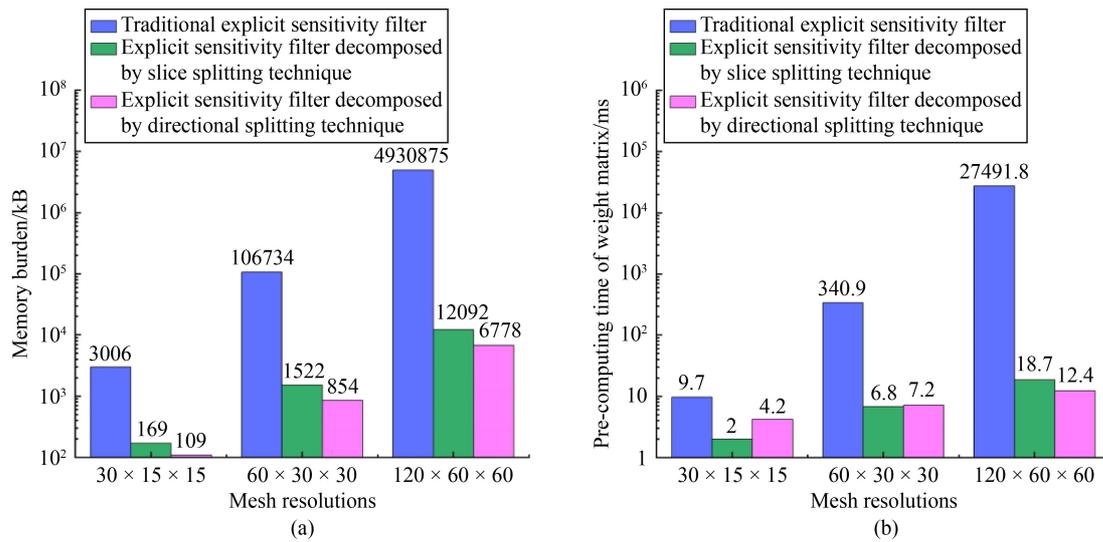


Fig. 29 Comparisons between the traditional explicit sensitivity filter and explicit sensitivity filters decomposed by slice and directional splitting technique for (a) the memory burden of weight matrix and (b) the pre-computing time of weight matrix.

According to the numerical results presented in Sections 4 and 5, the splitting techniques of tensor product structure are applied successfully to the implicit and explicit filters, and the decomposed implicit filters deliver exact identical converged designs and convergence history as they are generated by the original implicit filters. In addition, the decomposed explicit filters can be utilized to replace the traditional centroid distance-based explicit filter because the generated results are very close to each other. Despite the exact (or approximate) equivalence property in the optimization process, the performance in the efficiencies related to the sensitivity analysis and weight matrix between the traditional filters and the proposed decomposed filters are quite different. More concretely, when we replace the traditional filter with the decomposed filter proposed in this work, the average updating time of sensitivity analysis is decreased maximally by the factors of 55.8 and 1.72 for the design problems solved by IGA-based and FEM-based TO methods, respectively. For the two aspects of memory burden and pre-computing time for the filter weight matrix, the decomposed filter can reduce the memory burden by the factors 193733 and 759.5 and the pre-computing time by the factors 2701 and 2217 for IGA-based and FEM-based TO methods, respectively. Therefore, the proposed splitting technique of the tensor product structure can improve the efficiency of the implicit and explicit filters utilized in TO massively.

In the future, the proposed decomposed filters in this work will be extended to the dynamic [57–59] and additive manufacturing constrained design problems [60]. Moreover, the combination of the proposed filter with the geometric reconstruction-based IGA method [61] will be developed to extend the IGA-based TO method into design problems with complex CAD models.

Nomenclature

Abbreviations

ESO	Evolutionary structural optimization
FEM	Finite element method
IGA	Isogeometric analysis
MTOP	Multi-resolution topology optimization
NURBS	Non-uniform rational B-spline
PDE	Partial differential equation
SIMP	Solid isotropic material with penalization
TO	Topology optimization

Variables

c	Compliance
$\frac{\partial \tilde{c}}{\partial \bar{\rho}_{i,j}}$	Modified sensitivity of the (i,j) th design element
$d\mathbf{c}_{n_e \times n_e}$	Sensitivities of the objective function with respect to the centroid densities for all design elements
$d\mathbf{V}_{n_e \times n_e}$	Sensitivities of the volume fraction with respect to the centroid densities for all elements
E_0	Young's modulus of solid elements
E_{\min}	Young's modulus of void elements
$E(\rho_{ex}, \rho_{ey})$	Interpreted Young's modulus by the modified SIMP model of the (ex, ey) th IGA element
$E(\bar{\rho}_{i,j})$	Interpreted Young's modulus by the modified SIMP model of the (i,j) th finite element
\mathbf{f}	External load
$H_{i,j}$	Sum of the weights over the influence region of the (i,j) th design element

$\mathbf{H}_{m \times n e_x}(x^c)$	Univariate B-spline basis functions at the centroids of all design elements along with the x direction	$w_{(i,j),(i_1,j_1)}$	Weight of the sensitivity of the (i_1,j_1) th design element contributed to the modified sensitivity
$\mathbf{H}_{n_e \times n_e}(x), \mathbf{H}_{n_e \times n_e}(y)$	Weight matrices along with the x and y directions in FEM, respectively	(x_{ex}^c, y_{ey}^c)	Centroid coordinate of the (ex,ey) th IGA element
$\mathbf{H}_{n_e \times n_e}(y^c)$	Univariate B-spline basis functions at the centroids of all design elements along with the y direction	$\rho(x,y)$	Density of an arbitrary point (x,y) in the design domain
\mathbf{k}_0	Standard elemental stiffness matrix for the finite element mesh	$\rho_{i,j}$	Nodal density of the (i,j) th control points
$\mathbf{k}_{ex,ey}$	Solid elemental structural stiffness for the (ex,ey) th IGA element	$\rho_{ex,ey}^c$	Centroid density values for the (ex,ey) th IGA element
$\mathbf{K}(\rho)$	Global structural stiffness matrix in IGA	ρ	Union of nodal design variables
$\mathbf{K}(\bar{\rho})$	Global stiffness of the FEM mesh	$\bar{\rho}$	Union of discrete design variables in FEM
m, n	Number of control points along with the horizontal and vertical directions, respectively	γ	Parameter to avoid division by zero
$n e_x, n e_y, n e_z$	Number of elements along with the $x, y,$ and z directions, respectively	Ξ	Knot vector
$N_i^p(x)$	i th B-spline basis function along with the x direction with degree equals to p	Ψ	Admissible space for the union of nodal design variables ρ
$N_j^q(y)$	j th B-spline basis function along with the y direction with degree equals to q	$\bar{\Psi}$	Admissible space for $\bar{\rho}$
$N_{i,j}^{p,q}(x,y)$	Bivariate B-spline basis function with $(p \times q)$ -degree combination	$\theta_{\text{iterative_step}}$	Maximum derivation of design variables on consecutive iterative steps
$\bar{N}_{(i,j)}$	Neighborhood region of the (i,j) th element defined by the filter radius		
p	Degree of B-splines in the x direction		
$penal$	Penalty factor to steer black and white design		
q	Degree of B-splines in the y direction		
S	Overall storing burden of the B-spline weights		
S_x	Storing burden of the B-spline weights along with the x direction		
S_y	Storing burden of the B-spline weights along with the y direction		
$[r_{\min}]$	Filter radius		
$\mathbf{u}_{i,j}$	Local displacement vector for the (i,j) th finite element		
$\mathbf{u}_{ex,ey}$	Local displacement vector of control points for the (ex,ey) th IGA element		
$\mathbf{u}(\rho)$	Global displacement vector of control points		
$\mathbf{u}(\bar{\rho})$	Global displacement vector of the FEM mesh		
$volfrac$	Volume fraction of solid material		
V	Volume of solid material		
V_0	Volume of the whole design domain		
$w_{i,i}(x)$	Weight of the i_1 th design element along with the x direction contributed to the modified sensitivity		
$w_{j,i}(y)$	Weight of the j_1 th design element along with the y direction contributed to the modified sensitivity		

Acknowledgements This work has been supported by the National Key R&D Program of China (Grant No. 2020YFB1708300) and China Postdoctoral Science Foundation (Grant No. 2021M701310). No conflicts of interest exist in this paper. The authors would like to thank the anonymous reviewers, whose suggestions helped to substantially improve this manuscript.

References

1. Bendsoe M P, Sigmund O. Topology Optimization: Theory, Methods, and Applications. 2nd ed. Berlin: Springer, 2003
2. Bendsoe M P, Kikuchi N. Generating optimal topologies in structural design using a homogenization method. Computer Methods in Applied Mechanics and Engineering, 1988, 71(2): 197–224
3. Bendsoe M P, Sigmund O. Material interpolation schemes in topology optimization. Archive of Applied Mechanics, 1999, 69(9): 635–654
4. Sigmund O. A 99 line topology optimization code written in MATLAB. Structural and Multidisciplinary Optimization, 2001, 21(2): 120–127
5. Xie Y M, Steven G P. A simple evolutionary procedure for structural optimization. Computers & Structures, 1993, 49(5): 885–896
6. Allaire G, Jouve F, Toader A M. Structural optimization using sensitivity analysis and a level-set method. Journal of Computational Physics, 2004, 194(1): 363–393
7. Mei Y L, Wang X M. A level set method for structural topology optimization and its applications. Advances in Engineering Software, 2004, 35(7): 415–441
8. Guo X, Zhang W S, Zhong W L. Doing topology optimization explicitly and geometrically—a new moving morphable components based framework. Journal of Applied Mechanics, 2014, 81(8): 081009

9. Zhang W H, Zhou Y, Zhu J H. A comprehensive study of feature definitions with solids and voids for topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 2017, 325: 289–313
10. Zhang W S, Jiang S, Liu C, Li D D, Kang P, Youn S K, Guo X. Stress-related topology optimization of shell structures using IGA/TSA-based moving morphable void (MMV) approach. *Computer Methods in Applied Mechanics and Engineering*, 2020, 366: 113036
11. Zhou Y, Zhang W H, Zhu J H, Xu Z. Feature-driven topology optimization method with signed distance function. *Computer Methods in Applied Mechanics and Engineering*, 2016, 310: 1–32
12. Xie X D, Wang S T, Xu M M, Jiang N, Wang Y J. A hierarchical spline based isogeometric topology optimization using moving morphable components. *Computer Methods in Applied Mechanics and Engineering*, 2020, 360: 112696
13. Xie X D, Wang S T, Ye M, Xia Z H, Zhao W, Jiang N, Xu M M. Isogeometric topology optimization based on energy penalization for symmetric structure. *Frontiers of Mechanical Engineering*, 2020, 15(1): 100–122
14. Xie X D, Yang A D, Wang Y J, Jiang N, Wang S T. Fully adaptive isogeometric topology optimization using MMC based on truncated hierarchical B-splines. *Structural and Multidisciplinary Optimization*, 2021, 63(6): 2869–2887
15. Haber R B, Jog C S, Bendsøe M P. A new approach to variable-topology shape design using a constraint on perimeter. *Structural Optimization*, 1996, 11(1): 1–12
16. Borrvall T. Topology optimization of elastic continua using restriction. *Archives of Computational Methods in Engineering*, 2001, 8(4): 351–385
17. Bourdin B. Filters in topology optimization. *International Journal for Numerical Methods in Engineering*, 2001, 50(9): 2143–2158
18. Sigmund O. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, 2007, 33(4): 401–424
19. Andreassen E, Clausen A, Schevenels M, Lazarov B S, Sigmund O. Efficient topology optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization*, 2011, 43(1): 1–16
20. Bruns T E, Tortorelli D A. Topology optimization of non-linear elastic structures and compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering*, 2001, 190(26–27): 3443–3459
21. Sigmund O, Maute K. Sensitivity filtering from a continuum mechanics perspective. *Structural and Multidisciplinary Optimization*, 2012, 46(4): 471–475
22. Guest J K, Prévost J H, Belytschko T. Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *International Journal for Numerical Methods in Engineering*, 2004, 61(2): 238–254
23. Sigmund O. Manufacturing tolerant topology optimization. *Acta Mechanica Sinica*, 2009, 25(2): 227–239
24. Xu S L, Cai Y W, Cheng G D. Volume preserving nonlinear density filter based on heaviside functions. *Structural and Multidisciplinary Optimization*, 2010, 41(4): 495–505
25. Kawamoto A, Matsumori T, Yamasaki S, Nomura T, Kondoh T, Nishiwaki S. Heaviside projection based topology optimization by a PDE-filtered scalar function. *Structural and Multidisciplinary Optimization*, 2011, 44(1): 19–24
26. Lazarov B S, Sigmund O. Filters in topology optimization based on Helmholtz-type differential equations. *International Journal for Numerical Methods in Engineering*, 2011, 86(6): 765–781
27. Chen L L, Lu C, Lian H J, Liu Z W, Zhao W C, Li S Z, Chen H B, Bordas S P A. Acoustic topology optimization of sound absorbing materials directly from subdivision surfaces with isogeometric boundary element methods. *Computer Methods in Applied Mechanics and Engineering*, 2020, 362: 112806
28. Wallin M, Ivarsson N, Amir O, Tortorelli D. Consistent boundary conditions for PDE filter regularization in topology optimization. *Structural and Multidisciplinary Optimization*, 2020, 62(3): 1299–1311
29. Xie X D, Wang S T, Wang Y J, Jiang N, Zhao W, Xu M M. Truncated hierarchical B-spline-based topology optimization. *Structural and Multidisciplinary Optimization*, 2020, 62(1): 83–105
30. Xie X D, Yang A D, Jiang N, Wang S T. Topology optimization using fully adaptive truncated hierarchical B-splines. *Applied Mathematical Modelling*, 2021, 96: 131–151
31. Xie X D, Yang A D, Jiang N, Zhao W, Liang Z S, Wang S T. Adaptive topology optimization under suitably graded THB-spline refinement and coarsening. *International Journal for Numerical Methods in Engineering*, 2021, 122(20): 5971–5998
32. Qian X P. Topology optimization in B-spline space. *Computer Methods in Applied Mechanics and Engineering*, 2013, 265: 15–35
33. Wang M M, Qian X P. Efficient filtering in topology optimization via B-splines. In: *Proceedings of ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Buffalo: ASME, 2014, V02BT03A011
34. Costa G, Montemurro M, Pailhès J. A 2D topology optimisation algorithm in NURBS framework with geometric constraints. *International Journal of Mechanics and Materials in Design*, 2018, 14(4): 669–696
35. Costa G, Montemurro M, Pailhès J. NURBS hyper-surfaces for 3D topology optimization problems. *Mechanics of Advanced Materials and Structures*, 2021, 28(7): 665–684
36. Costa G, Montemurro M, Pailhès J. Minimum length scale control in a NURBS-based SIMP method. *Computer Methods in Applied Mechanics and Engineering*, 2019, 354: 963–989
37. Costa G, Montemurro M, Pailhès J, Perry N. Maximum length scale requirement in a topology optimisation method based on NURBS hyper-surfaces. *CIRP Annals*, 2019, 68(1): 153–156
38. Costa G, Montemurro M. Eigen-frequencies and harmonic responses in topology optimisation: a CAD-compatible algorithm. *Engineering Structures*, 2020, 214: 110602
39. Rodríguez T, Montemurro M, Le Texier P, Pailhès J. Structural displacement requirement in a topology optimization algorithm based on isogeometric entities. *Journal of Optimization Theory and Applications*, 2020, 184(1): 250–276
40. Montemurro M, Rodríguez T, Texier P L, Pailhès J. Multi-displacement requirement in a topology optimization algorithm based on non-uniform rational basis spline hyper-surfaces. In:

- Mariano P M, ed. *Variational Views in Mechanics*. Cham: Springer, 2021, 223–257
41. Roiné T, Montemurro M, Pailhès J. Stress-based topology optimization through non-uniform rational basis spline hyper-surfaces. *Mechanics of Advanced Materials and Structures*, 2021, In press
 42. Montemurro M, Bertolino G, Roiné T. A general multi-scale topology optimisation method for lightweight lattice structures obtained through additive manufacturing technology. *Composite Structures*, 2021, 258: 113360
 43. Bertolino G, Montemurro M. Two-scale topology optimisation of cellular materials under mixed boundary conditions. *International Journal of Mechanical Sciences*, 2022, 216: 106961
 44. Montemurro M, Refai K. A topology optimization method based on non-uniform rational basis spline hyper-surfaces for heat conduction problems. *Symmetry*, 2021, 13(5): 888
 45. Montemurro M, Refai K, Catapano A. Thermal design of graded architected cellular materials through a CAD-compatible topology optimisation method. *Composite Structures*, 2022, 280: 114862
 46. Montemurro M. On the structural stiffness maximisation of anisotropic continua under inhomogeneous Neumann–Dirichlet boundary conditions. *Composite Structures*, 2022, 287: 115289
 47. Fernandez F, Puso M A, Solberg J, Tortorelli D A. Topology optimization of multiple deformable bodies in contact with large deformations. *Computer Methods in Applied Mechanics and Engineering*, 2020, 371: 113288
 48. Wang C, Zhang W H, Zhou L, Gao T, Zhu J H. Topology optimization of self-supporting structures for additive manufacturing with B-spline parameterization. *Computer Methods in Applied Mechanics and Engineering*, 2021, 374: 113599
 49. Kolda T G, Bader B W. Tensor decompositions and applications. *SIAM Review*, 2009, 51(3): 455–500
 50. Sidiropoulos N D, De Lathauwer L, Fu X, Huang K J, Papalexakis E E, Faloutsos C. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 2017, 65(13): 3551–3582
 51. Castellana D, Bacciu D. A tensor framework for learning in structured domains. *Neurocomputing*, 2022, 470: 405–426
 52. Boussé M, Vervliet N, Domanov I, Debals O, De Lathauwer L. Linear systems with a canonical polyadic decomposition constrained solution: algorithms and applications. *Numerical Linear Algebra with Applications*, 2018, 25(6): e2190
 53. Scholz F, Mantzaflaris A, Jüttler B. Partial tensor decomposition for decoupling isogeometric Galerkin discretizations. *Computer Methods in Applied Mechanics and Engineering*, 2018, 336: 485–506
 54. Mantzaflaris A, Jüttler B, Khoromskij B N, Langer U. Low rank tensor methods in Galerkin-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 2017, 316: 1062–1085
 55. de Boor C. On calculating with B-splines. *Journal of Approximation Theory*, 1972, 6(1): 50–62
 56. Svanberg K. The method of moving asymptotes—a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 1987, 24(2): 359–373
 57. Xia Q, Zhou T, Wang M Y, Shi T L. Shape and topology optimization for tailoring the ratio between two flexural eigenfrequencies of atomic force microscopy cantilever probe. *Frontiers of Mechanical Engineering*, 2014, 9(1): 50–57
 58. Xu M M, Wang S T, Xie X D. Level set-based isogeometric topology optimization for maximizing fundamental eigenfrequency. *Frontiers of Mechanical Engineering*, 2019, 14(2): 222–234
 59. Long K, Yang X Y, Saeed N, Tian R H, Wen P, Wang X. Topology optimization of transient problem with maximum dynamic response constraint using SOAR scheme. *Frontiers of Mechanical Engineering*, 2021, 16(3): 593–606
 60. Liu J K, Chen Q, Liang X, To A C. Manufacturing cost constrained topology optimization for additive manufacturing. *Frontiers of Mechanical Engineering*, 2019, 14(2): 213–221
 61. Wang Y J, Gao L, Qu J P, Xia Z H, Deng X W. Isogeometric analysis based on geometric reconstruction models. *Frontiers of Mechanical Engineering*, 2021, 16(4): 782–797