Space–time isogeometric topology optimization with additive manufacturing constraints

Yang, Li; Wang, Weiming; Ji, Ye; Zhu, Chun-Gang; Wang, Charlie C.L.

DOI
10.1016/j.cma.2025.117976

Publication date
2025

Document Version
Final published version

Published in
Computer Methods in Applied Mechanics and Engineering

Computer Methods in Applied Mechanics and Engineering 441 (2025) 117976

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

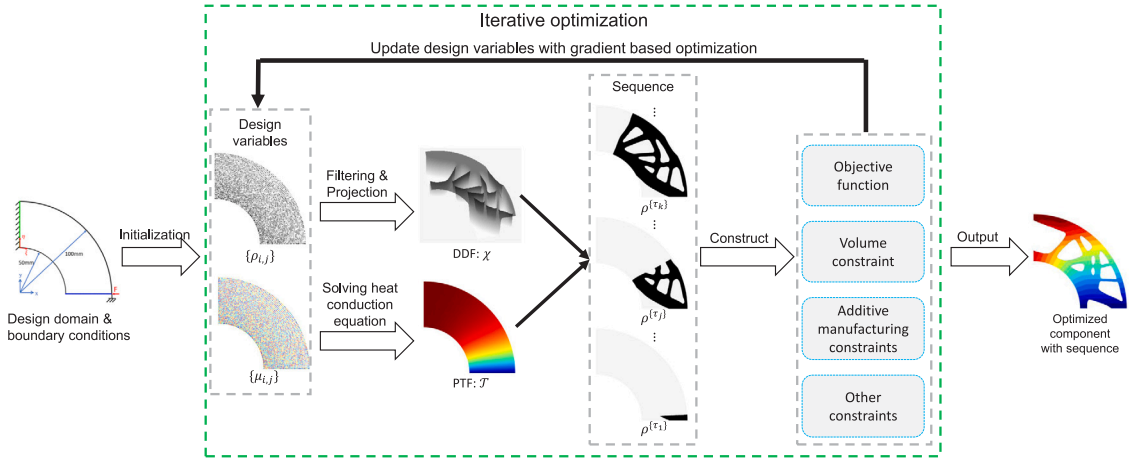
journal homepage: www.elsevier.com/locate/cma

Space–time isogeometric topology optimization with additive manufacturing constraints

Li Yang [a,b], Weiming Wang [b,*], Ye Ji [c], Chun-Gang Zhu [a], Charlie C.L. Wang [b]

[a] School of Mathematical Sciences, Dalian University of Technology, Dalian, China
[b] Department of Mechanical and Aerospace Engineering, The University of Manchester, Manchester, United Kingdom
[c] Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Isogeometric analysis
Space–time topology optimization
Heat conduction equation
Additive manufacturing
Layer thickness

ABSTRACT

This paper presents a novel space–time isogeometric topology optimization (ITO) framework for additive manufacturing, enabling concurrent optimization of structural shape and fabrication sequence with accurate geometric representation. The method integrates a density distribution function with a pseudo-time function to optimize build sequences for complex structures, with an objective function that minimizes compliance under external loads and accounts for self-weight effects during fabrication. Density values and virtual heat conduction coefficients are defined at B-spline control points to serve as design variables. A heat conduction-based formulation is employed to generate the pseudo-time function so that prevents the generation of isolated or floating material regions. A layer thickness constraint, defined by the pseudo-time gradient, further enhances manufacturability. The approach has been validated in 2D and 3D examples, demonstrating its effectiveness in managing objectives of entire structure's stiffness and self-weight of intermediate structures.

* Corresponding author.
E-mail address: wwmdlut@dlut.edu.cn (W. Wang).

https://doi.org/10.1016/j.cma.2025.117976
Received 20 November 2024; Received in revised form 22 February 2025; Accepted 25 March 2025
Available online 9 April 2025

1. Introduction

Topology optimization (TO) is a mathematical approach used in engineering design to determine the best material layout within a given design space for a given set of loads, boundary conditions, and constraints with the goal of maximizing the performance of the system. This method has been widely used in various fields, such as aerospace, automotive, civil engineering, and biomechanics [1]. Numerical analysis in the majority of current TO methods mainly employs the finite element method (FEM). Despite its widespread use, FEM has several limitations [2,3]: (1) the finite element mesh is unable to precisely represent the geometry of the structure, thereby reducing numerical accuracy; (2) the accuracy of the optimization results is influenced by the low-order (C^0) continuity of structural responses in FEM; (3) achieving a high-quality finite element mesh is challenging — especially in those cases with large deformation.

In recent years, the isogeometric analysis (IGA), introduced by Hughes and his colleagues [2,3], extends and generalizes conventional FEMs for numerical analysis. The core idea of IGA is to use the same basis functions (e.g., non-uniform rational b-splines (NURBS)) for both geometry parameterization and the finite-dimensional solution space in numerical analysis, ensuring consistency between the geometrical model and the numerical analysis model. IGA offers significant advantages over conventional FEM in several key aspects [3,4]: (1) structural analysis in IGA utilizes precise geometric representations rather than approximate finite element meshes; (2) refinement is easily implemented by re-indexing the parametric space, without additional communication with the initial geometric model; (3) the high-order continuity in IGA surpasses the limitations of conventional FEM, which generally only provides C^0 continuity between elements; (4) IGA can achieve higher computational accuracy than FEM with the same number of degrees of freedom.

The incorporation of IGA into TO, resulting in the creation of the isogeometric topology optimization (ITO) framework, has attracted considerable attention from researchers and engineers. This includes approaches such as density-based ITO methods [5–7], level-set-based ITO methods [8,9], and moving morphable components-based ITO design frameworks [10,11]. The associated ITO methods have also tackled various design challenges, demonstrating the effectiveness and efficiency of these approaches through optimized solutions. Wang et al. [12] proposed an ITO method for generating periodic lattice materials. To speed up the ITO process for graded lattice materials, the mechanical properties are expressed as a function of the unit cell's relative density, eliminating the need for iterative calculations during the optimization. Liu et al. [13] introduced an innovative framework that integrates IGA with global stress-constrained TO. By harnessing IGA's capability for precise geometric representation, this approach significantly improves computational efficiency and solution accuracy. It provides a foundational reference for incorporating stress constraints into TO. In addition, Gupta et al. [14] introduced an innovative adaptive ITO methodology based on the GIFT framework, utilizing PHT-splines for efficient mesh refinement. This approach significantly reduces the degrees of freedom and optimizes complex geometries by progressively refining coarse meshes, all while eliminating geometric errors. Their work serves as an important reference for improving computational efficiency in ITO. Karuthedath et al. [15] introduced a continuous density-field-based ITO method, leveraging PHT-splines for accurate density representation and adaptive mesh refinement. This approach not only reduces the degrees of freedom required during optimization but also improves manufacturability through a smoothing algorithm based on the Shepard function, offering a novel perspective for efficiently solving complex geometry optimization problems. In another study, Karuthedath et al. [16] proposed a computationally efficient methodology for TO of fourth-order plate structures within a multi-patch IGA framework, utilizing PHT-Splines. The approach combines C^1 continuous discretization of plate structures, C^0 continuous density fields, and an adaptive mesh refinement strategy to optimize geometries. Multiple numerical examples demonstrate significant improvements in solution accuracy, smooth topology, and computational efficiency. Xie et al. [17] presented a hierarchical spline-based ITO method, utilizing adaptive IGA with hierarchical B-splines to improve computational efficiency and accuracy. The method begins with a coarse mesh and refines it locally based on an error-based marking strategy, resulting in more accurate results and efficient optimization. Gao et al. [18] presented the multi-patch ITO method for designing periodic and graded cellular structures. By using Nitsche's method to couple non-conforming meshes across multiple NURBS patches, the approach allows efficient multi-patch analysis. Zhang et al. [19] presented a 3D shell structure TO method under stress constraints, combining the Moving Morphable Void approach with trimming technique-based IGA. Using NURBS and trimming curves, the method offers efficient design with fewer variables and high-accuracy stress analysis.

In this work, the method proposed by Gao et al. [6] is utilized to represent the design domain and optimize the material distribution. This method has demonstrated the superior effectiveness of IGA and ITO for addressing linear elastic problems. The optimizer aims to identify an optimized density distribution function that has sufficient smoothness and continuity to represent the structural topology, with its iso-contour/surface defining the structural boundaries. In IGA, the same NURBS basis functions of a design domain are employed to create the solution space as the density distribution function for the unknown structural responses in the analysis. During optimization, the density values at the control points are treated as design variables to be evolved. Although TO can be used to create high-performance structures, the optimized free-form shapes are often difficult to produce using traditional manufacturing methods. To overcome this challenge and materialize the outcomes of structural optimization, additive manufacturing (AM) techniques have recently been employed. However, structures generated by TO cannot automatically meet the fabrication requirements of AM. Therefore, extensive efforts have been made to bridge the gap between TO and AM. For instance, the overhang angle (e.g. [20,21]), infill structures (e.g. [22,23]), build orientation (e.g. [24,25]), scanning path planning (e.g. [26,27]), connectivity (e.g. [28,29]), and material anisotropy (e.g. [30,31]).

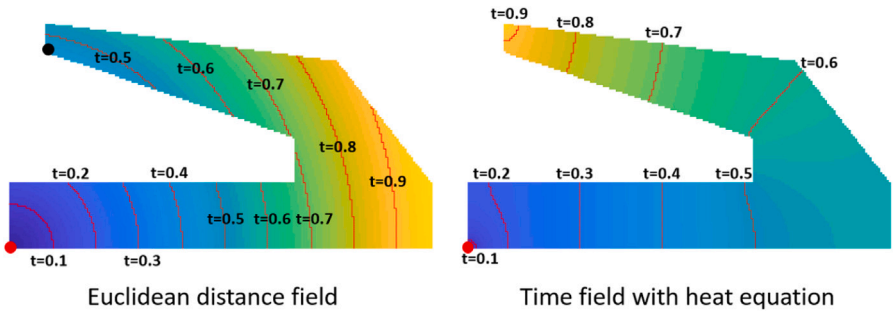


Fig. 1. Two different fields generated from the same starting point (red dot): (Left) an Euclidean distance field and (Right) the time field generated using heat equation. Nine isolines are plotted to indicate the sequences defined by the fields. Black dot shows a local minimum in the Euclidean distance field. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Although the above mentioned papers take fabrication factors into account, structural design and fabrication process are mainly dealt with separately, which may bring up some problems. For instance, in TO the mechanical property of the entire structure is mainly concerned, that is the compliance of the final structure. However, factors related to the manufacturing process, such as self-weight and process-dependent loads, which may significantly impact the quality and stability of fabrication, particularly when printing large structures, are often overlooked. These issues are closely intertwined with manufacturing process planning. In traditional AM technologies like fused deposition modeling (FDM) and selective laser sintering (SLS), which only allows for translation, the manufacturing processes are typically determined based on a given build orientation. In recent years, the emergence of multi-axis AM (e.g. [32–35]) using robotic systems cannot only provide translational motion but also reorient the print head or the component being constructed, introduces more printing degrees of freedom which increase the fabrication flexibility. With rotation, the fabrication sequence is not limited to planar layers. It allows to form structures with curved layers, which indicates that the manufacturing sequence can be optimized. Motivated by this, Wang et al. [36] proposed a space–time TO framework to concurrently optimize the structural layout and the fabrication sequence. In their method, a time field is applied to mimic the manufacturing process, in addition to a density field for defining the structural layout. The effectiveness of their method has been numerically demonstrated on multiple application scenarios, such as the self-weight of the intermediate structures and the robot weights during printing. AM process commonly involves high temperature gradient, resulting in large thermal distortion. In [37], Wang et al. proposed a method to minimize thermal-induced distortion through the manufacturing sequence in wire and arc AM, which is represented using a time field.

The time field is used to mimic the manufacturing process in both [36,37]. Each element in the design domain is associated with a time variable, where a smaller value indicates earlier production. Fabrication discontinuity, a crucial requirement for AM, can be characterized by local minima in the time field. This means material cannot be deposited at a location where its neighboring elements have not yet been produced. A minimum in time field is only permitted at a prescribed location where fabrication is supposed to start from (i.e. attached to the build plate). Although explicitly formulated constraints on time field introduced in their methods are able to avoid the appearance of local minima, their success strongly depends on the initialization of the time field. In previous studies [36,37], the initialization of the time field involved generating a Euclidean distance field from the starting points. However, for components with slightly complex geometry or topology, this approach may struggle to avoid local minima. For instance, in Fig. 1, the left field is generated based on the Euclidean distance. The left-bottom corner is specified as the starting point (i.e., the point with *zero* field value). Nine isolines (red curves) are shown which segment the entire shape to from the sequences. As can be observed from the figure, the Euclidean distance field has an extra local minima at the top-left (as indicated by the black dot).

It is well known that heat equation is able to generate continuous and monotonous virtual temperature field over a domain, which has been used to detect structural voids to meet the connectivity constraints in AM [38,39]. On the right of Fig. 1, we show a field generated using the heat equation, where the heat source is located in the bottom-left corner again and with insulated boundaries. To align with time field, we transform the computed virtual temperature field through heat equation into a time field. As we can observe that the field generated from the heat equation is free of local minima. Therefore, the virtual temperature field generated from the heat equation can be used to mimic the manufacturing process in our approach. In [40], Wang et al. integrated the heat conduction equation into the FEM-based TO method to generate the manufacturing sequence, which has demonstrated that the generated sequence is entirely free of local minima. Borrowing this idea, we formulate the heat conduction equation into our new framework of IGA-based TO for the problem of space–time concurrent optimization.

In our method, each control point is associated with two design variables: a density value that characterizes its material and a virtual heat conduction coefficient that governs the temperature distribution. Based on the virtual conduction coefficients at the control points, the steady-state heat conduction equation is solved to generate virtual temperature values at these points, which are then transformed into pseudo-time values indicating the manufacturing sequence. Based on the density values and pseudo-time values at the control points, along with the NURBS basis functions, a density distribution function (DDF) and a pseudo-time function (PTF) are constructed to continuously describe the material distribution and the pseudo build time within the design domain. The manufacturing sequences and volume constraints are defined based on these two continuous functions. To enhance

the manufacturability of the generated manufacturing sequence, a layer thickness constraint is established based on the gradient of the PTF. Unlike the method presented in [40], we do not multiply the density value by the virtual heat conduction coefficient at a control point to solve the steady-state heat conduction equation, which actually complicates the control of the gradient of the generated PTF.

The technical contribution of our approach can be summarized as:

- A novel framework that formulates the space–time concurrent optimization to generate the optimized structure and manufacturing sequence by IGA;
- A heat conduction based formation to resolve the floating region issues that can be generated by other field representations;
- Incorporating the manufacturing constraints such as self-weight prevention and layer-thickness control into the IGA-based TO.

Experimental results have been conducted on both 2D and 3D cases to demonstrate the effectiveness of the proposed method in managing the objectives of the entire structure's stiffness and the self-weight of intermediate structures.

In the AM process, previously printed structures may collapse or experience significant deformation due to their self-weight. Therefore, the effectiveness of our method has been numerically evaluated for minimizing the compliance resulted from the self-weight of the intermediate structures during printing in both 2D and 3D cases while using multi-axis additive manufacturing.

This paper is organized as follows: the topology description model using NURBS is presented in Section 2, including the basic knowledge of NURBS and the definition of DDF in Section 2.1 and numerical discretization in Section 2.2, respectively. In Section 3, the details of the proposed space–time isogeometric TO formulation will be introduced, including the generation of PTF in Section 3.1, the generation of differentiable layers in Section 3.2, the definition of the volume constraints for the intermediate structures in Section 3.3, the proposed optimization formulation in Section 3.4, and the sensitivity analysis in Section 3.5. The experimental results regarding self-weight for the 2D and 3D cases will be presented in Section 4.1 and Section 4.2, respectively. The results incorporating the layer thickness constraint will be discussed in Section 4.3. Several aspects of the proposed method will be discussed in Section 5, including the algorithm efficiency (Section 5.1), the influence of the basis functions' degree (Section 5.2), and the number of Gauss quadrature points (Section 5.3). The whole paper will end with the conclusion and the discussion of future work in Section 6.

2. Topology representation

Developing a method to depict the structural arrangement or constructing a model that accurately portrays material distribution poses a pivotal challenge in TO. IGA is a novel computational approach that offers the possibility of integrating the numerical analysis and computer-aided design into a single unified framework [8,9]. Various computational geometry technologies can be applied to IGA. In this work, the NURBS [41], which is the industry standard, is utilized to describe the design domain, as well as structural material distribution and manufacturing process.

Since the theoretical analysis and mathematical formulation of the 3D problem are highly consistent with those for 2D problem, we will only describe the theory and algorithm for the 2D case in the following sections. In this section, we will briefly introduce the basic knowledge of NURBS and the definition of DDF in Section 2.1 and numerical discretization of IGA in Section 2.2.

2.1. Geometric modeling and material description

For a given structure or design domain, its geometry is linearly approximated with NURBS basis functions using a specified set of control points $\mathbf{P} = \{\mathbf{P}_{i,j} | i = 1, \dots, n; j = 1, \dots, m\}$, with n and m representing the number of control points in two parametric directions, respectively, see the magenta points in Fig. 2(a). A NURBS surface $\mathbf{S}(\xi, \eta)$ using a tensor product is defined as:

$$\mathbf{S}(\xi, \eta) = \sum_{i=1}^{n} \sum_{j=1}^{m} R_{i,j}^{p,q}(\xi, \eta) \mathbf{P}_{i,j}, \tag{1}$$

where (ξ, η) is a point in parametric domain, p and q are the polynomial degrees of the basis functions, respectively, and $R_{i,j}^{p,q}(\xi, \eta)$ is a bivariate NURBS basis function defined in terms of the B-spline basis functions. The detailed definition of $R_{i,j}^{p,q}(\xi, \eta)$ can be found in the Appendix A.

The DDF was originally proposed in [6] to describe the material distribution. In this work, it is applied to describe the density field in the proposed space–time ITO framework. Therefore, we will briefly review its definition and core concept in this subsection. In the ITO, each control point $\mathbf{P}_{i,j}$ is assigned a scalar value $\rho_{i,j} \in [0, 1]$, representing its control density, which serves as the design variable in ITO. To avoid checkerboard issue caused in ITO process [1], the Shepard function [42] is applied to smooth the densities of the control points:

$$\tilde{\rho}_{i,j} = \sum_{i=1}^{\bar{n}} \sum_{j=1}^{\bar{m}} \psi(\rho_{i,j}) \rho_{i,j}, \tag{2}$$

where $\bar{n}$ and $\bar{m}$ denote the total number of control points situated within the local support area of the current control point in two parametric directions, respectively. ψ is the Shepard function, whose definition can be found in the Appendix B.

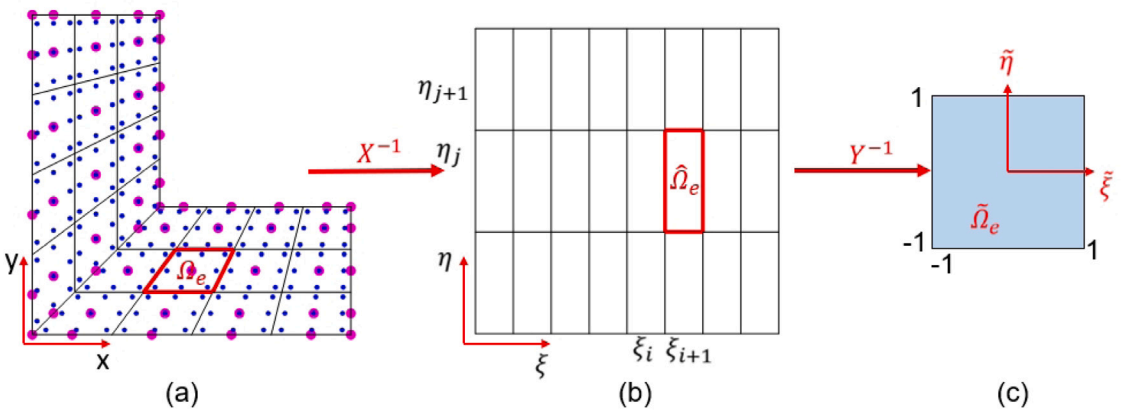


Fig. 2. The numerical integration of IGA elements. (a) illustrates an IGA mesh of an L-Shaped domain, where the blue points are Gauss quadrature points, the magenta points are control points, and Ω_e indicates an IGA element. (b) is the corresponding parametric space. (c) represents the bi-unit parent element, denoted as $\tilde{\Omega}_e$ which corresponds to the red element $\hat{\Omega}_e$ shown in (b). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

To ensure the densities of the control points converge to 1 (solid) or 0 (void), a commonly used Heaviside function [43] is employed to $\tilde{\rho}$, which is defined as:

$$\hat{\rho}_{i,j} = \frac{\tanh(\beta_d \sigma) + \tanh(\beta_d(\tilde{\rho}_{i,j} - \sigma))}{\tanh(\beta_d \sigma) + \tanh(\beta_d(1-\sigma))}, \tag{3}$$

where β_d controls the sharpness of the projection, and σ is the projection threshold, which is set to 0.5 in this work.

The DDF is defined by the linear combination of the projected densities $\hat{\rho}_{i,j}$ at the control points, weighted by the NURBS basis functions, which collectively describe the entire structural geometry and topology. The DDF $\mathcal{X}(\xi,\eta)$ in structural design domain is defined as:

$$\mathcal{X}(\rho,\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} R_{i,j}^{p,q}(\xi,\eta)\hat{\rho}_{i,j}. \tag{4}$$

The final structural boundaries of the optimized component are expressed by the iso-contour/iso-surface of the DDF.

2.2. Numerical discretization in IGA

For a linear elastic continuum, the static equilibrium equation is formulated as:

$$\mathbf{KU} = \mathbf{F}, \tag{5}$$

where $\mathbf{K}$ and $\mathbf{U}$ are the global stiffness matrix and displacement vector corresponding to the entire structure, respectively, and $\mathbf{F}$ denotes the external force. In the Galerkin IGA formulation [2], the global stiffness matrix and load vector are obtained by assembling the local stiffness matrices and load vectors, respectively. For a given design domain Ω, it is discretized into a set of IGA elements (see Ω_e in Fig. 2(a)). Then, the IGA elemental stiffness matrix can be calculated as:

$$\mathbf{K}_e = \int_{\Omega_e} \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B} d\Omega_e = \int_{\hat{\Omega}_e} \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}|J_1| d\hat{\Omega}_e = \int_{\tilde{\Omega}_e} \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}|J_1||J_2| d\tilde{\Omega}_e, \tag{6}$$

where $\mathbf{B}$ is the strain–displacement matrix calculated by partial derivatives of NURBS basis functions with respect to physical coordinates, and Ω_e, $\hat{\Omega}_e$, and $\tilde{\Omega}_e$ are the physical domain, parametric domain, and bi-unit parent element of the eth element ($e = 1, 2, \ldots, Nel$), with Nel representing the total number of IGA elements in the computational domain. As shown in Fig. 2, X: $\hat{\Omega}_e \rightarrow \Omega_e$ maps from the parametric space to the physical space, while Y: $\tilde{\Omega}_e \rightarrow \hat{\Omega}_e$ maps from the bi-unit parent element to the element in the parametric space. The integral for the calculation of IGA elemental stiffness matrix is first pulled back onto the parametric element and then onto the bi-unit parent element, involving the inverses of the two mappings. J_1 and J_2 are the Jacobi matrices of these two mappings, respectively.

The integrals defined in Eq. (6) are numerically solved using the Gauss quadrature method [44]. The accuracy of Gauss quadrature is influenced by the relationship between the number of quadrature points and the polynomial degree of the integrand. Specifically, we use the following rule, which is commonly adopted in the standard IGA, to determine the number of Gauss quadrature points: Number of Gauss quadrature points = Degree of basis function + 1. Consequently, Eq. (6) can be rewritten as:

$$\mathbf{K}_e = \sum_{i=1}^{p+1}\sum_{j=1}^{q+1} \left\{ \mathbf{B}^{\mathrm{T}}(\xi_{e_i},\eta_{e_j})(\mathcal{X}(\boldsymbol{\rho},\xi_{e_i},\eta_{e_j}))^{\gamma}\mathbf{D}_0\mathbf{B}(\xi_{e_i},\eta_{e_j})|J_1(\xi_{e_i},\eta_{e_j})||J_2(\xi_{e_i},\eta_{e_j})|\omega_{e_i}\omega_{e_j} \right\}, \tag{7}$$

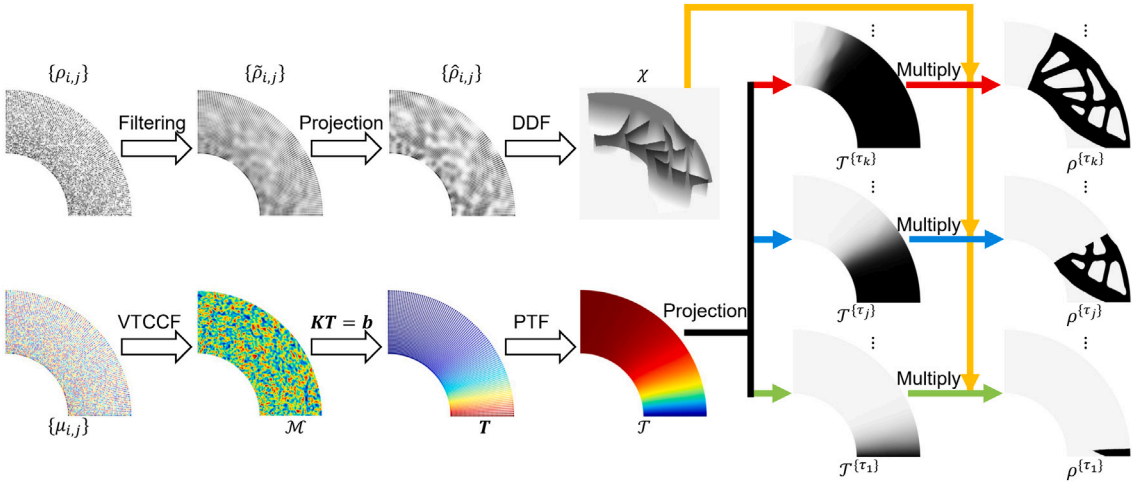


Fig. 3. The flowchart of the proposed space–time ITO framework.

where $\mathcal{X}(\rho, \xi_{e_i}, \eta_{e_j})$ represents the density at a Gauss quadrature point (ξ_{e_i}, η_{e_j}) of the eth IGA element, γ is the penalization parameter, ω_{e_i} and ω_{e_j} are the corresponding quadrature weights, respectively, and $\mathbf{D_0}$ is the elastic tensor matrix for the solid density. For demonstration, 3×3 Gauss quadrature points are shown for each IGA element in Fig. 2(a), represented as the blue points.

3. Space–time concurrent optimization using ITO

The flowchart of the proposed space–time isogeometric TO framework is presented in Fig. 3. The input design domain Ω is first parametrized using $m \times n$ control points and their corresponding NURBS basis functions, where the parametric domain is discretized into Nel IGA elements. The boundary and loading conditions are determined on the control points. Each control point $\mathbf{P}_{i,j}$ is assigned with two scalar values: a density value $\rho_{i,j} \in [0, 1]$ and a virtual thermal conductivity coefficient $\mu_{i,j} \in [0, 1]$. They serve as the design variables in the proposed framework. The density values are filtered to $\{\tilde{\rho}_{i,j}\}$, and then projected to be black or white $\{\hat{\rho}_{i,j}\}$. The DDF $\mathcal{X}$ and the virtual thermal conductivity coefficient function (VTCCF) $\mathcal{M}$ are generated using $\{\hat{\rho}_{i,j}\}$ and $\boldsymbol{\mu} = \{\mu_{i,j}\}$, respectively. The virtual temperature values $\boldsymbol{T} = \{T_{i,j}\}$ at the control points are obtained by solving the steady state heat conduction equation with $\mathcal{M}$, which are transformed into the pseudo-time values $\boldsymbol{t} = \{t_{i,j}\}$ and the PTF $\mathcal{T}$, which will be described in detail in Section 3.1. The projected $\mathcal{T}$ at the prescribed threshold τ is multiplied by $\mathcal{X}$ to form an intermediate structure $\rho^{\{\tau\}}$, which will be described in detail in Section 3.2. The proposed optimization framework, presented in Section 3.4, is solved using a gradient-based solver, specifically the method of moving asymptotes (MMA) [45]. The sensitivities of the objective functions and constraints will be introduced in Section 3.5.

3.1. Generation of PTF

Similar to the method described in [40], the steady state heat conduction equation with a drain term is used to calculate the PTF. The details will be briefly reviewed in the following. The steady state heat conduction equation is defined as:

$$\nabla(\mu(x, y)\nabla T(x, y)) - \delta_T T(x, y) = 0, \tag{8}$$

where ∇ denotes the vector differential operator, $T(x, y)$ is the function of temperature, $\mu(x, y)$ is the function of thermal conductivity coefficient which determines the distribution of the temperature, and δ_T is a parameter used to scale the drain term, which is defined as:

$$\delta_T = \beta_T / l_c^2, \tag{9}$$

where β_T is a given non-dimensional constant, and l_c is the characteristic length of the design domain.

Eq. (8) can be discretized on Ω with IGA as:

$$\mathbf{K}_T \mathbf{T} = \mathbf{b}, \tag{10}$$

where $\mathbf{T}$ represents the virtual temperatures of the control points, $\mathbf{b}$ is the given thermal load, and $\mathbf{K}_T$ is the global heat conductivity matrix, which is assembled from elemental matrix:

$$\mathbf{K}_T = \sum_e \mathbf{K}_T^e + \sum_e \mathbf{K}_C^e. \tag{11}$$

In Eq. (11), the first element matrix $\mathbf{K}_T^e$ is associated with the thermal diffusivity, which can be calculated as:

$$\mathbf{K}_T^e = \int_{\Omega_e} \mu(x,y)\mathbf{B}_e^{\mathrm{T}}\mathbf{B}_e d\Omega_e, \tag{12}$$

where $\mathbf{B}_e$ is the shape function matrix of the eth element, which is calculated as follows:

$$\mathbf{B}_e = \begin{bmatrix} \frac{\partial R_{i_1}}{\partial x} & \frac{\partial R_{i_2}}{\partial x} & \cdots & \frac{\partial R_{i_{(p+1)(q+1)}}}{\partial x} \\ \frac{\partial R_{i_1}}{\partial y} & \frac{\partial R_{i_2}}{\partial y} & \cdots & \frac{\partial R_{i_{(p+1)(q+1)}}}{\partial y} \end{bmatrix}, \tag{13}$$

where $R_{i_1}, R_{i_2}, \ldots, R_{i_{(p+1)(q+1)}}$ represent the NURBS basis functions that are non-zero over this element.

The second element matrix $\mathbf{K}_C^e$ is associated with the drain term, which is calculated as:

$$\mathbf{K}_C^e = \int_{\Omega_e} \delta_T \hat{R}^T \hat{R} d\Omega_e. \tag{14}$$

where $\hat{R} = [R_{i_1}, R_{i_2}, \ldots, R_{i_{(p+1)(q+1)}}]$. Instead of directly calculating $\mathbf{K}_C^e$ in Eq. (14) using NURBS basis functions, a lumped matrix is adopted for simplification by assuming that all the mass is lumped at the control points [40]:

$$\mathbf{K}_C^e = \delta_T * I, \tag{15}$$

where I is the identity diagonal matrix.

Similar to the Eq. (4), the VTCCF is defined using the virtual thermal conductivity coefficients at the control points as follows:

$$\mathcal{M}(\boldsymbol{\mu},\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} R_{i,j}^{p,q}(\xi,\eta)\mu_{i,j}. \tag{16}$$

Unlike the method proposed by Wang et al. [40], our approach does not involve multiplying the virtual thermal conductivity coefficient by the corresponding density value for each control point. Then, the integral defined in Eq. (12) can be computed using the Gauss quadrature method as:

$$\mathbf{K}_T^e = \sum_{l=1}^{p+1}\sum_{k=1}^{q+1}\left\{\mathcal{M}(\boldsymbol{\mu},\xi_{e_l},\eta_{e_k})\mathbf{B}_e^{\mathrm{T}}(\xi_{e_l},\eta_{e_k})\mathbf{B}_e(\xi_{e_l},\eta_{e_k})|J_1(\xi_{e_l},\eta_{e_k})||J_2(\xi_{e_l},\eta_{e_k})|\omega_{e_l}\omega_{e_k}\right\}, \tag{17}$$

where $\mathcal{M}(\boldsymbol{\mu},\xi_{e_l},\eta_{e_k})$ is the thermal conductivity coefficient at a Gauss quadrature point (ξ_{e_l},η_{e_k}) of the eth IGA element.

After solving the Eq. (10), the virtual temperature values $\{T_{i,j}\}$ for all control points are obtained. Then, a pseudo-time for the control point $\mathbf{P}_{i,j}$ is calculated as $t_{i,j} = 1 - T_{i,j}$. Subsequently, the PTF is defined to describe the pseudo-time of the whole design domain as follows:

$$\mathcal{T}(\boldsymbol{\mu},\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} R_{i,j}^{p,q}(\xi,\eta)t_{i,j}. \tag{18}$$

3.2. Generation of differentiable layers

At a specific time point τ, the already printed structure is denoted as $\{\mathcal{X}(\boldsymbol{\rho},\xi,\eta)|\mathcal{T}(\boldsymbol{\mu},\xi,\eta) \le \tau\}$, where $\mathcal{T}(\boldsymbol{\mu},\xi,\eta)$ is the pseudo-time at a point (ξ,η), and $\mathcal{X}(\rho,\xi,\eta)$ is the projected density value at this point. To ensure the differentiability of the above process, a Heaviside projection function is first used to convert the time values of control points smaller (or larger) than the threshold τ, to close to 1 (or 0). The projected time value at a control point is defined as:

$$\bar{t}_{i,j}^{\{\tau\}} = \frac{\tanh(\beta_t(1-\tau)) - \tanh(\beta_t(t_{i,j}-\tau))}{\tanh(\beta_t\tau) + \tanh(\beta_t(1-\tau))}, \tag{19}$$

where β_t controls the projection sharpness. Then, the projected PTF at time value τ is calculated as follows:

$$\mathcal{T}^{\{\tau\}}(\boldsymbol{\mu},\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} R_{i,j}^{p,q}(\xi,\eta)\bar{t}_{i,j}^{\{\tau\}}. \tag{20}$$

Consequently, the density function of the intermediate structure at the time value τ is defined as follows:

$$\rho^{\{\tau\}}(\boldsymbol{\rho},\boldsymbol{\mu},\xi,\eta) = \mathcal{X}(\boldsymbol{\rho},\xi,\eta) * \mathcal{T}^{\{\tau\}}(\boldsymbol{\mu},\xi,\eta). \tag{21}$$

With the definition of the intermediate structure, we are able to generate differentiable layers. First, the range of the pseudo printing time [0, 1] is uniformly divided into l intervals, denoted as $[\tau_{i-1},\tau_i]$, where $i = 1,\ldots,l$, with $\tau_0 = 0$ and $\tau_i = \frac{i}{l}$. The ith layer is written as $L_i = \{\mathcal{X}(\boldsymbol{\rho},\xi,\eta)|\mathcal{T}(\boldsymbol{\mu},\xi,\eta) \in [\tau_{i-1},\tau_i)\}$, and the last layer is $L_l = \{\mathcal{X}(\boldsymbol{\rho},\xi,\eta)|\mathcal{T}(\boldsymbol{\mu},\xi,\eta) \in [\tau_{l-1},1]\}$. This discrete layered decomposition can then be used in process simulation of AM. To facilitate gradient-based numerical optimization, the density function of the ith layer can be calculated using the functions of the intermediate structures as follows:

$$\triangle\,\rho^{\{\tau_i\}}(\boldsymbol{\rho},\boldsymbol{\mu},\xi,\mu) = \begin{cases} \mathcal{X}(\boldsymbol{\rho},\xi,\mu) - \rho^{\{\tau_{i-1}\}}(\boldsymbol{\rho},\boldsymbol{\mu},\xi,\eta), & i = l, \\ \rho^{\{\tau_i\}}(\boldsymbol{\rho},\boldsymbol{\mu},\xi,\eta) - \rho^{\{\tau_{i-1}\}}(\boldsymbol{\rho},\boldsymbol{\mu},\xi,\eta), & i = 2,\ldots,l-1, \\ \rho^{\{\tau_i\}}(\boldsymbol{\rho},\boldsymbol{\mu},\xi,\eta), & i = 1. \end{cases} \tag{22}$$

Since the self-weight of intermediate structures will be used to evaluate the proposed framework, Eq. (22) will be applied to calculate the equivalent self-weight loads for each layer. It is important to note that the intermediate structures and layers defined in this work are represented as continuous functions.

3.3. Volume constraints on intermediate structures

In AM, structures are fabricated layer upon layer on the top of each other. Fabrication speed is an important process parameter in AM, i.e., the amount of material which can be deposited per unit of time. In this work, we assume the fabrication speed is a constant for simplicity, that is the material deposited in each uniform time interval is almost the same. In other words, the addition of structure volume in each layer is restrained. The intermediate structures defined in Eq. (21) are continuous functions. The volume constraint for the ith intermediate structure is defined as follows:

$$-\epsilon \leq g_i(\boldsymbol{\rho},\boldsymbol{\mu}) = \frac{1}{|\Omega|}\int_{\Omega} \rho^{\{\tau\}}(\boldsymbol{\rho},\boldsymbol{\mu},\xi,\eta)\upsilon_0(\xi,\mu)d\Omega - \frac{i}{l}V_m \leq 0, \quad i = 1,\dots,l, \tag{23}$$

where V_m is the maximum available volume, $\upsilon_0(\xi,\mu)$ is the volume at the point (ξ,μ), and ϵ is a parameter to relax the constraint. To calculate the volume of each intermediate structure in a discrete manner and ensure high computational accuracy, Eq. (23) can be numerically solved using the Gauss quadrature method.

3.4. Formulation

Based on the definitions and constraints introduced in the previous subsections, the proposed space–time isogeometric TO formulation for the self-weight of intermediate structures is given as:

$$\begin{aligned}
\underset{\boldsymbol{\rho},\boldsymbol{\mu}}{min} \quad & c = \mathbf{U}^{\mathsf{T}}(\boldsymbol{\rho})\mathbf{K}(\boldsymbol{\rho})\mathbf{U}(\boldsymbol{\rho}) + \sum_{i=1}^{l}\alpha_i\left(\mathbf{U}^{\{i\}}(\boldsymbol{\rho},\boldsymbol{\mu})\right)^{\mathsf{T}}\mathbf{K}^{\{i\}}(\boldsymbol{\rho},\boldsymbol{\mu})\mathbf{U}^{\{i\}}(\boldsymbol{\rho},\boldsymbol{\mu}),\\
s.t. \quad & \mathbf{K}(\boldsymbol{\rho})\mathbf{U}(\boldsymbol{\rho}) = \mathbf{F},\\
& \mathbf{K}^{\{i\}}(\boldsymbol{\rho},\boldsymbol{\mu})\mathbf{U}^{\{i\}}(\boldsymbol{\rho},\boldsymbol{\mu}) = \mathbf{G}^{\{i\}}(\boldsymbol{\rho},\boldsymbol{\mu}), i = 1,\dots,l,\\
& \mathbf{K}_T(\boldsymbol{\mu})\mathbf{T}(\boldsymbol{\mu}) = \mathbf{b},\\
& \frac{1}{|\Omega|}\int_{\Omega}\mathcal{X}(\boldsymbol{\rho},\xi,\eta)\upsilon_0(\xi,\eta)d\Omega - V_m \leq 0,\\
& -\epsilon \leq g_i(\boldsymbol{\rho},\boldsymbol{\mu}) - \frac{i}{l}V_m \leq 0, i = 1,\dots,l,\\
& 0 \leq \rho_{i,j} \leq 1, i = 1,2,\dots,n; j = 1,2,\dots,m,\\
& 0 \leq \mu_{i,j} \leq 1, i = 1,2,\dots,n; j = 1,2,\dots,m,
\end{aligned} \tag{24}$$

where the objective function c consists of two components: the compliance of the entire structure due to external loads and the sum of the compliances of intermediate structures resulting from self-weight. α_i is a weighting factor that balances the two components of the objective function and is assigned the same value for all intermediate structures. The first two constraints are the governing mechanical equations of the entire structure and intermediate structures, respectively. These mechanical equations are independent of each other, and thus can be solved in parallel. $\mathbf{K}^{\{i\}}$, $\mathbf{U}^{\{i\}}$, and $\mathbf{G}^{\{i\}}$ represent the stiffness matrix, displacement vector, and equivalent self-weight loads for the ith intermediate structure, respectively. $\mathbf{K}^{\{i\}}$ is constructed using the density function of the ith intermediate structure. The calculation of $\mathbf{G}^{\{i\}}$ can be found in the article [36]. The third constraint is the heat conduction equation. The fourth and fifth constraints are the volume constraints of entire structure and all intermediate structures. υ_0 is the volume fraction of solid material. Both the intermediate structures and the entire structure are analyzed using IGA method. A gradient-based approach, the method of moving asymptotes (MMA) [45], is employed to solve the above formulation.

3.5. Sensitivity analysis

The adjoint analysis method is employed to compute the derivatives of the objective function and constraints with respect to the design variables. Firstly, we will introduce the details of the derivative of the objective function with respect to the virtual heat conduction coefficient. The Lagrangian function corresponding to the objective function c is formulated as follows:

$$L_o = c + \lambda^{\mathsf{T}}\left(\mathbf{K}_T(\boldsymbol{\mu})\mathbf{T}(\boldsymbol{\mu}) - \mathbf{b}\right), \tag{25}$$

where λ is the Lagrangian multiplier vector. Let $\mathcal{M}(\boldsymbol{\mu},\xi_{e_l},\eta_{e_k}) = \mathcal{M}_{e_{l,k}}$, and $\mathcal{T}(\boldsymbol{\mu},\xi_{e_l},\eta_{e_k}) = \mathcal{T}_{e_{l,k}}$. Then, the derivative of L_o with respect to virtual heat conduction coefficient $\mu_{i,j}$ is given by:

$$\begin{aligned}
\frac{\partial L_o}{\partial \mu_{i,j}} &= \frac{\partial c}{\partial \mu_{i,j}} + \lambda^{\mathsf{T}}\left(\frac{\mathbf{K}_T}{\partial \mu_{i,j}}\mathbf{T} + \mathbf{K}_T\frac{\partial \mathbf{T}}{\partial \mu_{i,j}}\right),\\
&= \sum_{e=1}^{Nel}\sum_{l=1}^{p+1}\sum_{k=1}^{q+1}\left(\frac{\partial c}{\partial \mathcal{T}_{e_{l,k}}}\sum_{\bar{i}=1}^{n}\sum_{\bar{j}=1}^{m}\frac{\partial \mathcal{T}_{e_{l,k}}}{\partial t_{\bar{i},\bar{j}}}\frac{\partial t_{\bar{i},\bar{j}}}{\partial \mathcal{M}_{e_{l,k}}} + \lambda^{\mathsf{T}}\left(\frac{\partial \mathbf{K}_T}{\partial \mathcal{M}_{e_{l,k}}}\mathbf{T} + \mathbf{K}_T\frac{\partial \mathbf{T}}{\partial \mathcal{M}_{e_{l,k}}}\right)\right)\frac{\partial \mathcal{M}_{e_{l,k}}}{\partial \mu_{i,j}}.
\end{aligned} \tag{26}$$

Since $t_{\bar{i},\bar{j}} = 1 - T_{\bar{i},\bar{j}}$, then,

$$\frac{\partial L_o}{\partial \mu_{i,j}} = \sum_{e=1}^{Nel}\sum_{l=1}^{p+1}\sum_{k=1}^{q+1}\left(-\frac{\partial c}{\partial \mathcal{T}_{e_{l,k}}}\sum_{\bar{i}=1}^{n}\sum_{\bar{j}=1}^{m}\frac{\partial \mathcal{T}_{e_{l,k}}}{\partial t_{\bar{i},\bar{j}}}\frac{\partial T_{\bar{i},\bar{j}}}{\partial \mathcal{M}_{e_{l,k}}} + \lambda^{\mathsf{T}}\left(\frac{\partial \mathbf{K}_T}{\partial \mathcal{M}_{e_{l,k}}}\mathbf{T} + \mathbf{K}_T\frac{\partial \mathbf{T}}{\partial \mathcal{M}_{e_{l,k}}}\right)\right)\frac{\partial \mathcal{M}_{e_{l,k}}}{\partial \mu_{i,j}}. \tag{27}$$

Let $M = -\frac{\partial c}{\partial \mathcal{T}_{e_{l,k}}}\left[\frac{\mathcal{T}_{e_{l,k}}}{\partial t_{1,1}}, \frac{\mathcal{T}_{e_{l,k}}}{\partial t_{2,1}}, \ldots, \frac{\mathcal{T}_{e_{l,k}}}{\partial t_{m,n}}\right]^{\mathsf{T}}$, and $\frac{\partial \mathbf{T}}{\partial \mathcal{M}_{e_{l,k}}} = \left[\frac{\partial T_{1,1}}{\partial \mathcal{M}_{e_{l,k}}}, \frac{\partial T_{2,1}}{\partial \mathcal{M}_{e_{l,k}}}, \ldots, \frac{\partial T_{m,n}}{\partial \mathcal{M}_{e_{l,k}}}\right]^{\mathsf{T}}$, where $\frac{\partial c}{\partial \mathcal{T}_{e_{l,k}}}$ can be calculated using the method proposed in article [40], $\frac{\partial T_{i,j}}{\partial \mathcal{M}_{e_{l,k}}}$ can be calculated using Eqs. (17) and (10), and $\frac{\mathcal{T}_{e_{l,k}}}{\partial t_{i,j}} = R_{i,j}^{p,q}(\xi_{e_l}, \eta_{e_k})$. Then,

$$\begin{aligned}\frac{\partial L_o}{\partial \mu_{i,j}} &= \sum_{e=1}^{Nel}\sum_{l=1}^{p+1}\sum_{k=1}^{q+1}\left(M^{\mathsf{T}}\frac{\partial \mathbf{T}}{\partial \mathcal{M}_{e_{l,k}}} + \lambda^{\mathsf{T}}\left(\frac{\partial \mathbf{K}_T}{\partial \mathcal{M}_{e_{l,k}}}\mathbf{T} + \mathbf{K}_T\frac{\partial \mathbf{T}}{\partial \mathcal{M}_{e_{l,k}}}\right)\right)\frac{\partial \mathcal{M}_{e_{l,k}}}{\partial \mu_{i,j}},\\ &= \sum_{e=1}^{Nel}\sum_{l=1}^{p+1}\sum_{k=1}^{q+1}\left(\left(M^{\mathsf{T}} + \lambda^{\mathsf{T}}\mathbf{K}_T\right)\frac{\partial \mathbf{T}}{\partial \mathcal{M}_{e_{l,k}}} + \lambda^{\mathsf{T}}\frac{\partial \mathbf{K}_T}{\partial \mathcal{M}_{e_{l,k}}}\mathbf{T}\right)\frac{\partial \mathcal{M}_{e_{l,k}}}{\partial \mu_{i,j}},\end{aligned} \tag{28}$$

where λ can be solved with the following equation:

$$M^{\mathsf{T}} + \lambda^{\mathsf{T}}\mathbf{K}_T = 0. \tag{29}$$

Finally, the derivative of L_o regarding to $\mu_{i,j}$ is calculated as follows:

$$\frac{\partial L_o}{\partial \mu_{i,j}} = \sum_{e=1}^{Nel}\sum_{l=1}^{p+1}\sum_{k=1}^{q+1}\lambda^{\mathsf{T}}\frac{\partial \mathbf{K}_T}{\partial \mathcal{M}_{e_{l,k}}}\mathbf{T}\frac{\partial \mathcal{M}_{e_{l,k}}}{\partial \mu_{i,j}}, \tag{30}$$

where $\frac{\partial \mathbf{K}_T}{\partial \mathcal{M}_{e_{l,k}}}$ can be calculated with Eqs. (17) and (16), and $\frac{\partial \mathcal{M}_{e_{l,k}}}{\partial \mu_{i,j}} = R_{i,j}^{p,q}(\xi_{e_l}, \eta_{e_k})$.

The sensitivity of the objective function c with respect to the density design variable $\rho_{i,j}$ at the control point $\mathbf{P}_{i,j}$ can be calculated using the method described in the above paragraphs. which is given as:

$$\frac{\partial L_o}{\partial \rho_{i,j}} = \sum_{e=1}^{Nel}\sum_{l=1}^{p+1}\sum_{k=1}^{q+1}\frac{\partial c}{\partial \mathcal{X}(\boldsymbol{\rho}, \xi_{e_l}, \eta_{e_k})}\frac{\partial \mathcal{X}(\boldsymbol{\rho}, \xi_{e_l}, \eta_{e_k})}{\partial \hat{\rho}_{i,j}}\frac{\partial \hat{\rho}_{i,j}}{\partial \tilde{\rho}_{i,j}}\frac{\partial \tilde{\rho}_{i,j}}{\partial \rho_{i,j}}, \tag{31}$$

where $\frac{\partial c}{\partial \mathcal{X}(\boldsymbol{\rho}, \xi_{e_l}, \eta_{e_k})}$ can be calculated using the method proposed in article [40]. $\frac{\partial \mathcal{X}(\boldsymbol{\rho}, \xi_{e_l}, \eta_{e_k})}{\partial \hat{\rho}_{i,j}}$, $\frac{\partial \hat{\rho}_{i,j}}{\partial \tilde{\rho}_{i,j}}$, and $\frac{\partial \tilde{\rho}_{i,j}}{\partial \rho_{i,j}}$ can be calculated using Eqs. (4), (3), and (2), respectively.

For the volume constraints on intermediate structures Eq. (23), we also use the adjoint sensitivity analysis method to calculate their sensitivities. The Lagrangian function for the volume constraint on the sth intermediate structure, $g_s(\boldsymbol{\rho}, \boldsymbol{\mu})$, is given as follows:

$$L_c = g_s(\boldsymbol{\rho}, \boldsymbol{\mu}) - \frac{s}{l}V_m + \boldsymbol{\beta}^{\mathsf{T}}(\mathbf{K}_T\mathbf{T} - \mathbf{b}), \tag{32}$$

where $\boldsymbol{\beta}$ is the Lagrangian multiplier vector. Then, the derivative of L_c with respect to design variable $\mu_{i,j}$ is calculated as:

$$\begin{aligned}\frac{\partial L_c}{\partial \mu_{i,j}} &= \frac{\partial g_s}{\partial \mu_{i,j}} + \boldsymbol{\beta}^{\mathsf{T}}\left(\frac{\mathbf{K}_T}{\partial \mu_{i,j}}\mathbf{T} + \mathbf{K}_T\frac{\partial \mathbf{T}}{\partial \mu_{i,j}}\right),\\ &= \sum_{e=1}^{Nel}\sum_{l=1}^{p+1}\sum_{k=1}^{q+1}\left(\frac{\partial g_s}{\partial \mathcal{T}_{e_{l,k}}}\sum_{\bar{i}=1}^{n}\sum_{\bar{j}=1}^{m}\frac{\partial \mathcal{T}_{e_{l,k}}}{\partial t_{\bar{i},\bar{j}}}\frac{\partial t_{\bar{i},\bar{j}}}{\partial \mathcal{M}_{e_{l,k}}} + \boldsymbol{\beta}^{\mathsf{T}}\left(\frac{\partial \mathbf{K}_T}{\partial \mathcal{M}_{e_{l,k}}}\mathbf{T} + \mathbf{K}_T\frac{\partial \mathbf{T}}{\partial \mathcal{M}_{e_{l,k}}}\right)\right)\frac{\partial \mathcal{M}_{e_{l,k}}}{\partial \mu_{i,j}},\\ &= \sum_{e=1}^{Nel}\sum_{l=1}^{p+1}\sum_{k=1}^{q+1}\left(-\frac{\partial g_s}{\partial \mathcal{T}_{e_{l,k}}}\sum_{\bar{i}=1}^{n}\sum_{\bar{j}=1}^{m}\frac{\partial \mathcal{T}_{e_{l,k}}}{\partial t_{\bar{i},\bar{j}}}\frac{\partial T_{\bar{i},\bar{j}}}{\partial \mathcal{M}_{e_{l,k}}} + \boldsymbol{\beta}^{\mathsf{T}}\left(\frac{\partial \mathbf{K}_T}{\partial \mathcal{M}_{e_{l,k}}}\mathbf{T} + \mathbf{K}_T\frac{\partial \mathbf{T}}{\partial \mathcal{M}_{e_{l,k}}}\right)\right)\frac{\partial \mathcal{M}_{e_{l,k}}}{\partial \mu_{i,j}}.\end{aligned} \tag{33}$$

Let $Q = -\frac{\partial g_s}{\partial \mathcal{T}_{e_{l,k}}}\left[\frac{\partial \mathcal{T}_{e_{l,k}}}{\partial t_{1,1}}, \frac{\partial \mathcal{T}_{e_{l,k}}}{\partial t_{2,1}}, \ldots, \frac{\partial \mathcal{T}_{e_{l,k}}}{\partial t_{m,n}}\right]^{\mathsf{T}}$, then

$$\begin{aligned}\frac{\partial L_c}{\partial \mu_{i,j}} &= \sum_{e=1}^{Nel}\sum_{l=1}^{p+1}\sum_{k=1}^{q+1}\left(Q^{\mathsf{T}}\frac{\partial \mathbf{T}}{\partial \mathcal{M}_{e_{l,k}}} + \boldsymbol{\beta}^{\mathsf{T}}\left(\frac{\partial \mathbf{K}_T}{\partial \mathcal{M}_{e_{l,k}}}\mathbf{T} + \mathbf{K}_T\frac{\partial \mathbf{T}}{\partial \mathcal{M}_{e_{l,k}}}\right)\right)\frac{\partial \mathcal{M}_{e_{l,k}}}{\partial \mu_{i,j}},\\ &= \sum_{e=1}^{Nel}\sum_{l=1}^{p+1}\sum_{k=1}^{q+1}\left(\left(Q^{\mathsf{T}} + \boldsymbol{\beta}^{\mathsf{T}}\mathbf{K}_T\right)\frac{\partial \mathbf{T}}{\partial \mathcal{M}_{e_{l,k}}} + \boldsymbol{\beta}^{\mathsf{T}}\frac{\partial \mathbf{K}_T}{\partial \mathcal{M}_{e_{l,k}}}\mathbf{T}\right)\frac{\partial \mathcal{M}_{e_{l,k}}}{\partial \mu_{i,j}},\end{aligned} \tag{34}$$

where $\boldsymbol{\beta}$ can be solved with the following equation:

$$Q^{\mathsf{T}} + \boldsymbol{\beta}^{\mathsf{T}}\mathbf{K}_T = 0. \tag{35}$$

Then,

$$\frac{\partial L_c}{\partial \mu_{i,j}} = \sum_{e=1}^{Nel}\sum_{l=1}^{p+1}\sum_{k=1}^{q+1}\boldsymbol{\beta}^{\mathsf{T}}\frac{\partial \mathbf{K}_T}{\partial \mathcal{M}_{e_{l,k}}}\mathbf{T}\frac{\partial \mathcal{M}_{e_{l,k}}}{\partial \mu_{i,j}}. \tag{36}$$

The derivative of L_c with respect to design variable $\rho_{i,j}$ is calculated as:

$$\frac{\partial L_c}{\partial \rho_{i,j}} = \sum_{e=1}^{Nel} \sum_{l=1}^{p+1} \sum_{k=1}^{q+1} \frac{\partial g_s}{\partial \mathcal{X}(\boldsymbol{\rho}, \xi_{e_l}, \eta_{e_k})} \frac{\partial \mathcal{X}(\boldsymbol{\rho}, \xi_{e_l}, \eta_{e_k})}{\partial \hat{\rho}_{i,j}} \frac{\partial \hat{\rho}_{i,j}}{\partial \tilde{\rho}_{i,j}} \frac{\partial \tilde{\rho}_{i,j}}{\partial \rho_{i,j}}. \tag{37}$$

It is worth noting that, Eq. (32) only includes one part of the volume constraint for the sth intermediate structure. The derivative of the other part of the constraint $g_s(\boldsymbol{\rho}, \boldsymbol{\mu})$ with respect to both $\rho_{i,j}$ and $\mu_{i,j}$ can be calculated in a similar manner to the approach described above.

Algorithm 1: Space–Time ITO Algorithm

Input: NURBS geometry; global volume fraction $V_{\max}$; initial variables ρ, μ; $nloop$; ϵ_0;
Output: Optimized variables ρ, μ.

Control points design variable ρ, μ;
Iteration index $i = 0$;
Projection parameter $\beta_d = 1$, $\beta_t = 10$;
cold = 0, cnew = 1;
while $i \leq nloop$ *and* $|cold - cnew| > \epsilon_0$ **do**
  $i = i + 1$;
  $\tilde{\rho} \leftarrow \rho$ via the Shepard function Eq. (2);
  $\hat{\rho} \leftarrow \tilde{\rho}$ via Heaviside function Eq. (3);
  Compute DDF, VTCCF;
  Compute global heat conductivity K_T and virtual temperature vector T via Eq. (10)-(17);
  Compute pseudo-time via Eq. (18);
  Compute global stiffness matrix K and global displacement vector U via Eq. (5);
  Objective function c and sensitivity analysis $\frac{\partial L_o}{\partial \mu_{i,j}}$, $\frac{\partial L_o}{\partial \rho_{i,j}}$ via Eq. (30)-(31);
  $cold \leftarrow cnew$, $cnew \leftarrow c$;
  Global volume constraints and sensitivity analysis;
  Volume constraint of the intermediate structures and sensitivity analysis $\frac{\partial L_c}{\partial \mu_{i,j}}$, $\frac{\partial L_c}{\partial \rho_{i,j}}$ via Eq. (36)-(37);
  Layer thickness constraint via Eq. (38) and sensitivity analysis via Eq. (39)-(40);
  Update ρ, μ using MMA solver;
  if $mod(i, 20) == 0$ *and* $\beta_d < 7$ **then**
    $\beta_d = \beta_d + 2$;
  if $mod(i, 30) == 0$ *and* $\beta_t < 130$ **then**
    $\beta_t = \beta_t + 30$;
Return the final design topology;

4. Experimental results

In this section, we will first assess the effectiveness of the proposed space–time ITO formulation through a 2D case study in Section 4.1. We will analyze the impact of different weighting factors α_i in the objective function on the performance of intermediate structures and the convergence of the proposed method. Subsequently, our method will be tested on 3D cases to further demonstrate its effectiveness in Section 4.2. To make the generated sequence theoretically manufacturable, an extra constraint is proposed to ensure the thickness of each layer is approximately uniform. The proposed constraint and the corresponding results will be presented in Section 4.3. The efficiency of the proposed algorithm, the influences of the degree of the basis functions and the number of Gauss quadrature points on the performance of the proposed method will be discussed in Section 5.

In all experiments, the Young's modulus is fixed at 1.0, and the Poisson's ratio is fixed at 0.3. The penalization parameter γ used in Eq. (7) is set to 3. Unless otherwise specified, the maximum available volume fraction is set to 0.6, with the number of layers set to 20 for 2D cases and 10 for 3D cases, respectively. The value of β_T defined in Eq. (9) is set to 0.01. The Heaviside projection parameter β_d for the densities of the control points, as defined in Eq. (3), starts at 1 and is increased by 2 every 20 iterations until it reaches 7. Similarly, the parameter β_t for the pseudo-time of control points, as defined in Eq. (19), starts at 10 and is increased by 30 every 30 iterations until it reaches 130. The optimization process is terminated if the difference between consecutive objective function values is smaller than a specified threshold ϵ_0 (set to 10^{-3} in this work) or if the maximum number of iterations (e.g., 500) is reached. The parameterization details for the simple domains tested in this section and the parameterization method used for the complex domain are provided in Appendix C. For simplicity, we will not describe the parameterization information for each example in the following subsections. The pseudo-code of the proposed space–time ITO algorithm is presented in Alg. 1. Our method is implemented with Matlab on a PC with AMD Ryzen 9 5900X 12-Core Processor @ 3.70 GHz, 64 GB memory, and NVIDIA GeForce RTX 2070 SUPER GPU.

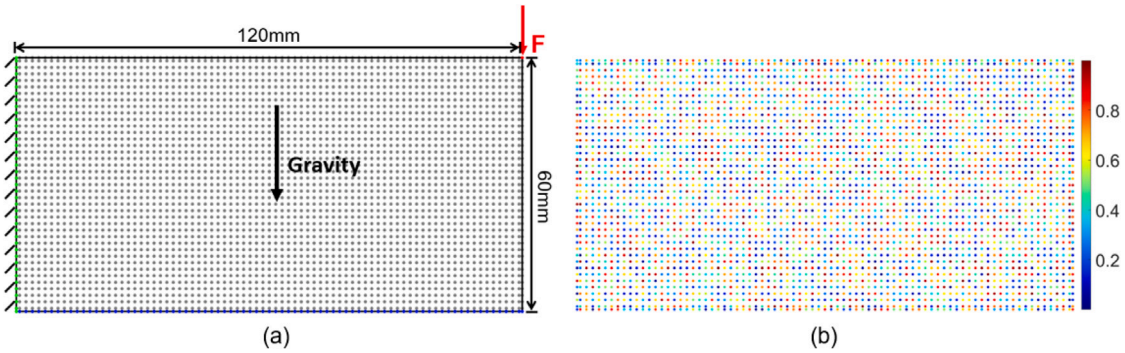


Fig. 4. Structural geometry of the 2D Cantilever Beam with boundary and loading conditions: (a) control points, where the blue points at the bottom indicate the heat source, green points on the left are fixed, and the red point at the top right represents the location with force applied; (b) initial heat conductivity coefficient distribution. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

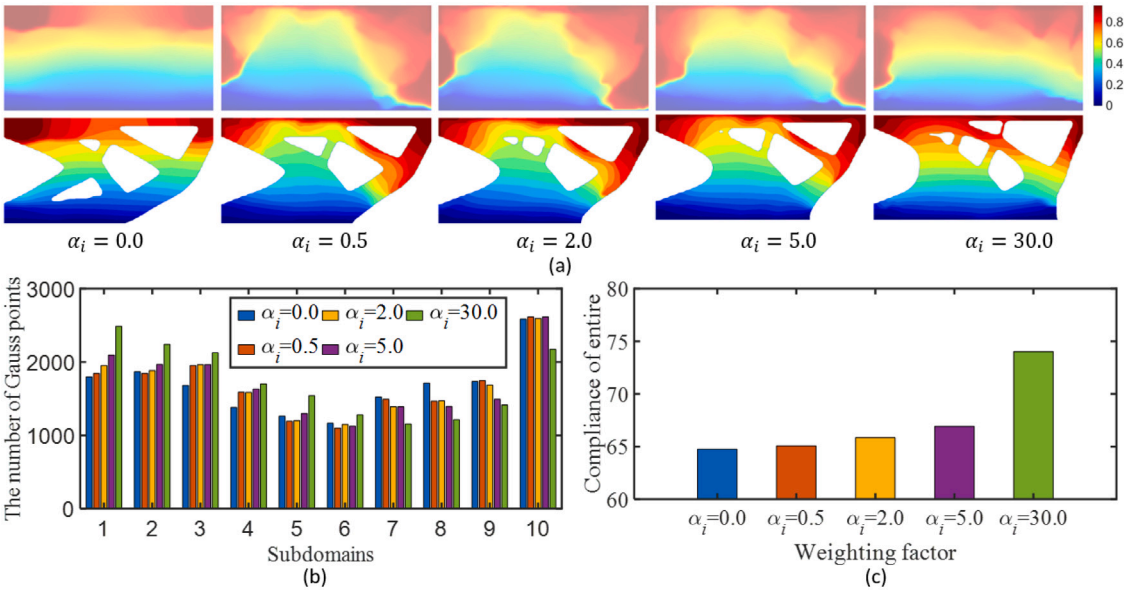


Fig. 5. Experimental results on the 2D Cantilever beam with varying values of the weights α_i. The optimized time fields and components, along with the manufacturing sequences, are shown in (a), where the values of α_i are 0.0, 0.5, 2.0, 5.0, and 30.0 from left to right. The variations in the solid Gauss points along the y-axis direction are visualized in (b). The compliances of the components are displayed in (c). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.1. 2D case study

To demonstrate the effectiveness of the proposed framework (24), we test it on the design domain shown in Fig. 4(a), with dimensions of 120 mm × 60 mm. The design domain is approximated using a NURBS surface with control points arranged in a dimension of 81 × 41, see the black points in Fig. 4(a). The control points on the left boundary are fixed, and a global downward force F with a magnitude of 1.0 is applied at the right-top corner control point (the red arrow). The black arrow indicates the direction of gravity. The gravity is uniformly distributed across the control points, including all boundary control points, to calculate the equivalent self-weight loading, ensuring that the total self-weight loading sums to 1. The control points on the bottom boundary (the blue control points) are fixed when solving the governing equations for intermediate structures, as defined in Eq. (10). The initial virtual heat conductive coefficients $\boldsymbol{\mu}$ for all control points are randomly generated, which are shown in Fig. 4(b). In the figure, the red color represents large values, while the blue color indicates small values.

For the objective function defined in Eq. (24), the values of all α_i are set to be identical. In this experiment, five different values of α_i are evaluated, which are 0.0, 0.5, 2.0, 5.0 and 30.0. The optimized results are shown in Fig. 5. The first row of Fig. 5(a) illustrates the optimized pseudo-time fields, while the second row depicts the optimized structures along with the fabrication sequences. In the pseudo-time fields, dark blue represents regions that will be fabricated first, while dark red indicates regions that will be fabricated last. In the second row of Fig. 5(a), different colors represent various fabrication layers. From Fig. 5(a), we can

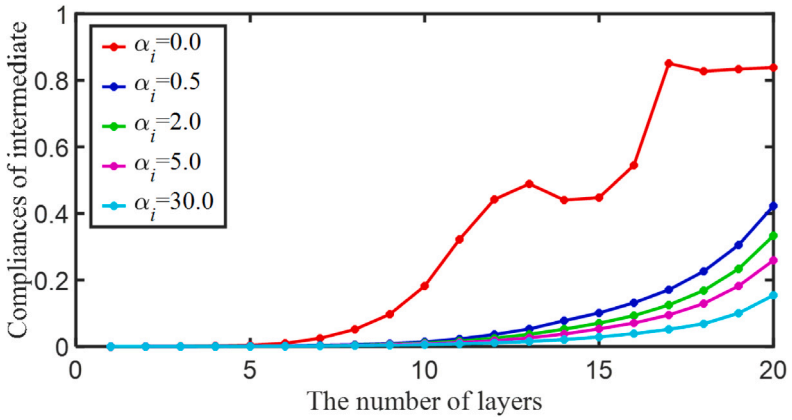


Fig. 6. Compliances of intermediate structures for different weighting factors. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

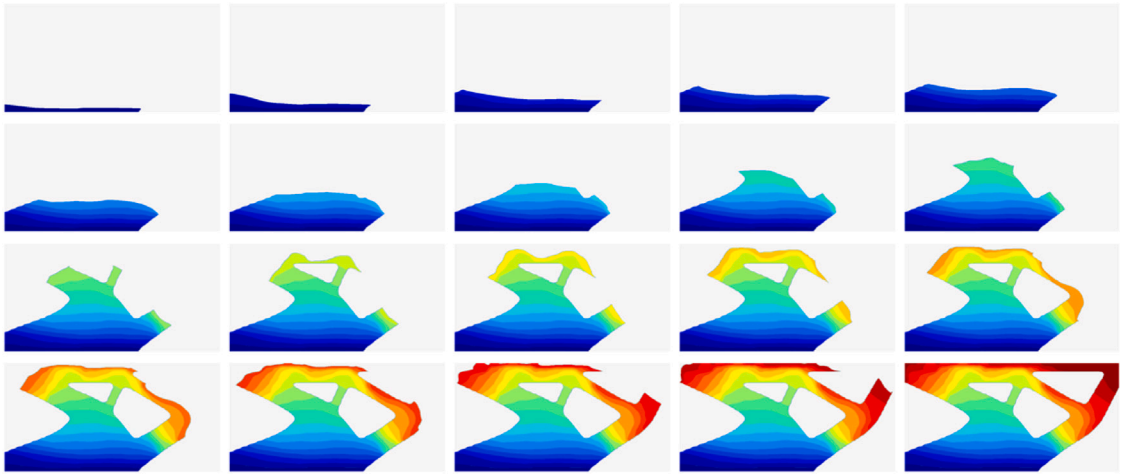


Fig. 7. Visualization of the intermediate structures in the manufacturing sequence generated using $\alpha_i = 0.5$.

observe that the optimized pseudo-time fields are continuous and free of local minima. As the value of α_i increases, the bottom region in the optimized component progressively widens. This occurs because the bottom boundary is fixed to account for the self-weight of intermediate structures. To visualize the movement of the solid material, we uniformly divide the design domain into ten subregions along vertical direction. We then count the number of Gauss points that have solid material within each subregion. The statistical results are plotted in Fig. 5(b). From this figure, we can clearly see that the structure generated using $\alpha_i = 30.0$ has the largest number of Gauss point with solid material in the first six subregions, while having the smallest number of Gauss points in the last four subregions. In contrast, the structure generated using $\alpha_i = 0.0$ shows the smallest number of Gauss points with solid material in the first four subregions, and the relatively large number of Gauss points in the last four subregions. The compliances of the entire structures are shown in Fig. 5(c). It demonstrates that as the value of α_i increases, the compliance also increases. Specifically, it rises from 64.75 for $\alpha_i = 0.0$ to 74.02 for $\alpha_i = 30.0$. However, the increase remains within an acceptable range.

The compliances of intermediate structures resulting from the self-weight for different weighting factors are plotted in Fig. 6. The red curve, representing $\alpha_i = 0.0$, shows the largest compliance at each intermediate structure. As the value of α_i increases, the compliances at all intermediate structures decrease. For example, the compliance at the 15th intermediate structure is decreased from 0.45 for $\alpha_i = 0.0$ to 0.03 for $\alpha_i = 30.0$, resulting in a reduction of 93.33%. This demonstrates that the structural performance of intermediate structures under self-weight is improved following optimization. The layers corresponding to $\alpha_i = 0.5$ are plotted in Fig. 7, illustrating the manufacturing process for each layer. The convergence curves of the objective function and global volume constraint are plotted on the left-hand side of Fig. 8, while the convergence curves for the volume constraints of the intermediate structures are plotted on the right-hand side of Fig. 8. For simplicity, we only plot the convergence curves of the volume constraints for the odd-numbered intermediate structures. From these curves, we can observe that the objective function rapidly decreases and stabilizes after 150 iterations, indicating that it nearly converges by that point. Additionally, all volume constraints are satisfied after 150 iterations.

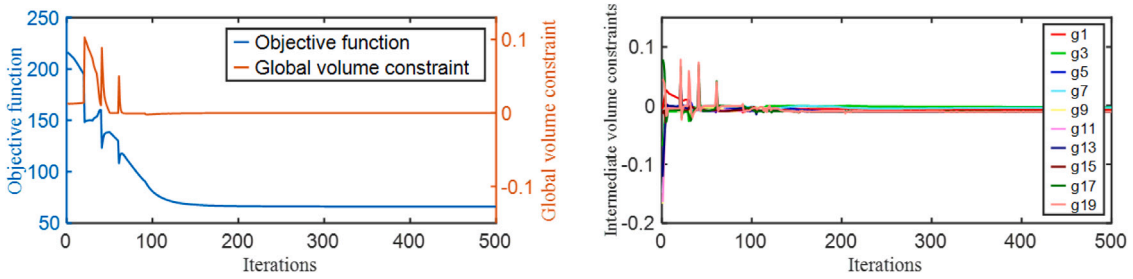


Fig. 8. The convergence curves of an optimized 2D structure corresponding to $\alpha_i = 0.5$. Left: the objective function and global volume constraint. Right: the volume constraints for the odd-numbered intermediate structures.

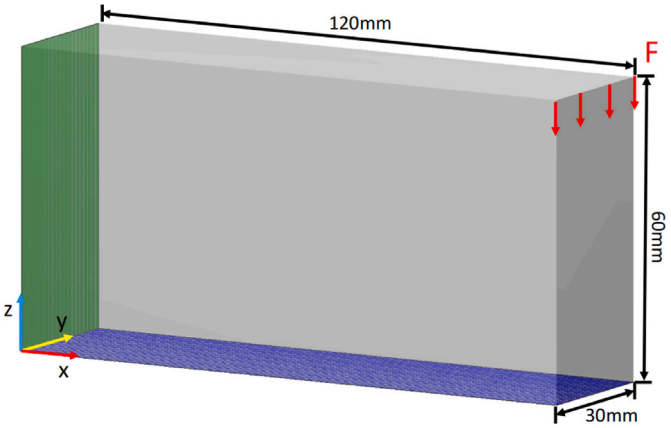


Fig. 9. The design domain, dimensions, heat source, and boundary/loading conditions for the 3D Cantilever beam. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.2. 3D case studies

In this section, our method has been tested on two 3D cases to further demonstrate its effectiveness. The design domain, dimensions, and boundary/loading conditions (the red arrows) of the first 3D case are shown in Fig. 9. In this setup, the control points on the left surface (the green surface) are fixed, and a distributed load is applied to the control points on the top-right edge. When solving the governing equation for the intermediate structures, the control points on the bottom surface (the blue surface) are fixed. The direction of the gravity is aligned with the negative z-axis.

Similar to the 2D case, five values of α_i in the objective function are investigated, including 0.00, 0.01, 0.05, 0.10 and 0.15. The optimized virtual time fields are shown in the top row of Fig. 10(a), while the optimized components along with their manufacturing sequences are plotted in the bottom row of Fig. 10(a). It can be observed that all virtual time fields are smooth and free of local minima. Due to the NURBS representation, the surface of the optimized component can be drawn with greater smoothness compared to that optimized using the voxel representation. As the value of α_i increases, the top bar in the optimized component becomes thinner, while the bottom region becomes wider. To visualize the movement of solid material, we divide the design domain into ten subregions along the z-axis, and count the number of Gauss points with solid material within each subregion. The statistics are plotted in Fig. 10(b). From this figure, we can see that as the value of α_i increases, more solid material is shifted towards the bottom (as indicated by the bar charts for the first two subregions and the last two subregions). This redistribution occurs because the control points on the bottom surface are fixed.

The compliances corresponding to the components in the bottom row of Fig. 10(a) are plotted in Fig. 10(c). Similarly, as the value of α_i increases, the compliance of the entire component increases. The compliances of all intermediate structures for different weighting factors are plotted in Fig. 11. For each intermediate structure, its compliance decreases as the value of α_i increases. For instance, the compliance of the 9th intermediate structure decreases from 364.38 for $\alpha_i = 0.0$ to 24.18 for $\alpha_i = 0.10$, representing a reduction of 87.87%. The intermediate structures corresponding to the component generated using $\alpha_i = 0.01$ are illustrated in Fig. 12. From this figure, we can clearly see that the generated manufacturing sequence has no overhang parts, indicating that the optimized virtual time field is smooth and free of local minima. The convergence curves of the objective function and global volume constraint are plotted on the left-hand side of Fig. 13, while the convergence curves for the volume constraints of the intermediate structures are plotted on the right-hand side of Fig. 13. From these curves, we can observe that the objective function rapidly decreases and stabilizes after 200 iterations, indicating that it nearly converges by that point. Additionally, all volume constraints are well satisfied.

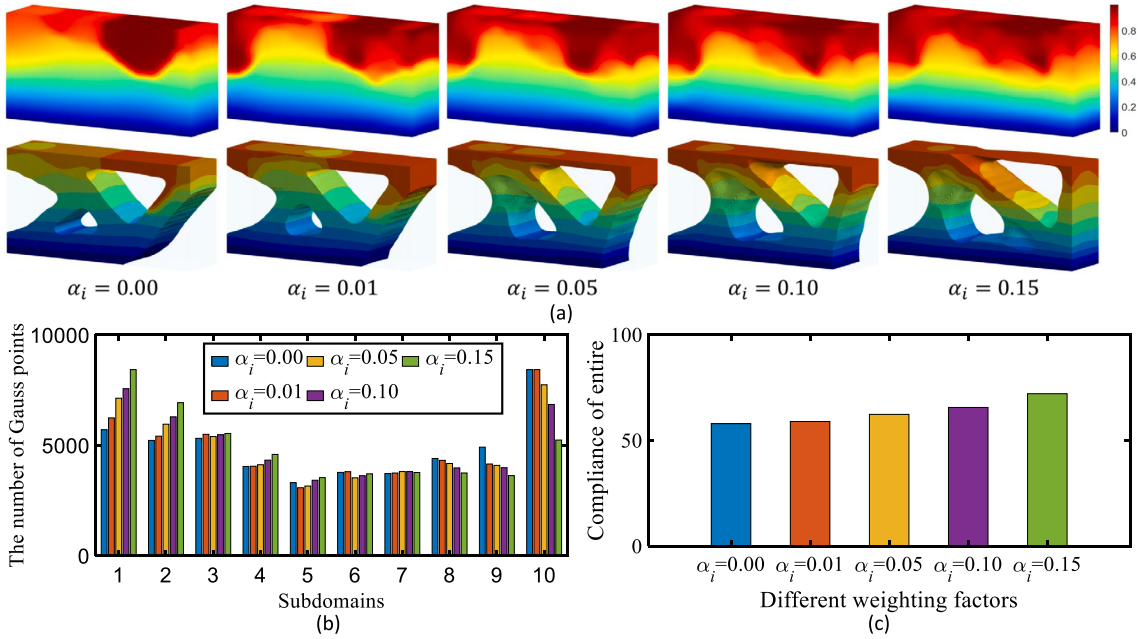


Fig. 10. Experimental results on the 3D Cantilever beam with varying values of the weighting factor α_i. The optimized time fields and components, along with the manufacturing sequences, are shown in (a), where the values of α_i are 0.0, 0.01, 0.05, 0.1, and 0.15 from left to right. The variations in the solid Gauss points along the z-axis direction are visualized in (b). The compliances of the components are displayed in (c).

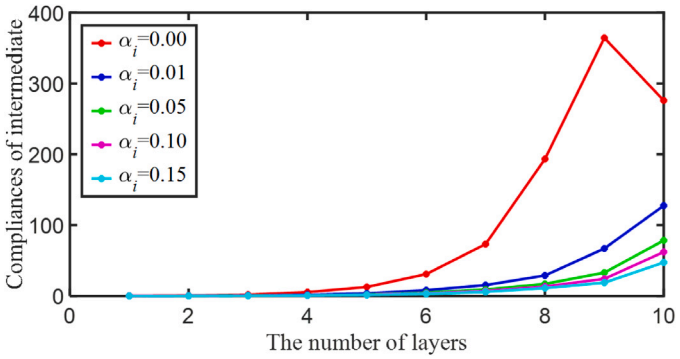


Fig. 11. Compliances of intermediate structures for various values of weighting factor α_i.

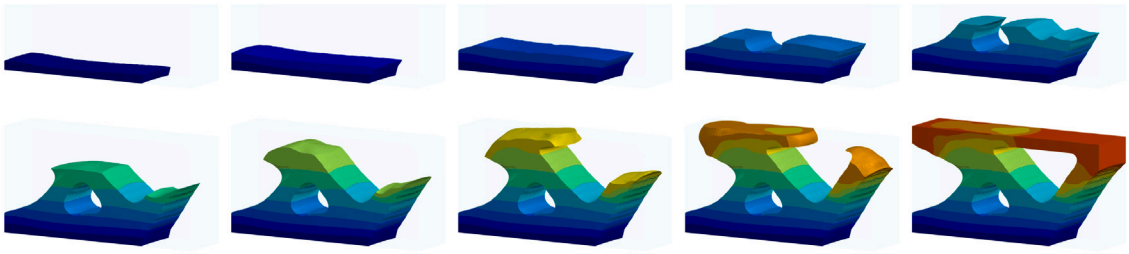


Fig. 12. Visualization of the intermediate structures in the manufacturing sequence generated with $\alpha = 0.01$.

The dimensions, control points (the blue points), and boundary/loading conditions of the 3D Bone design domain are shown in Fig. 14. In this setup, the control points on the bottom surface (represented by the green points) are fixed to solve the governing equations of the global component and the intermediate structure. These control points also act as heat sources for solving the heat conduction equation. Two loads are applied to the left-top and right-top control point (represented by the red arrows).

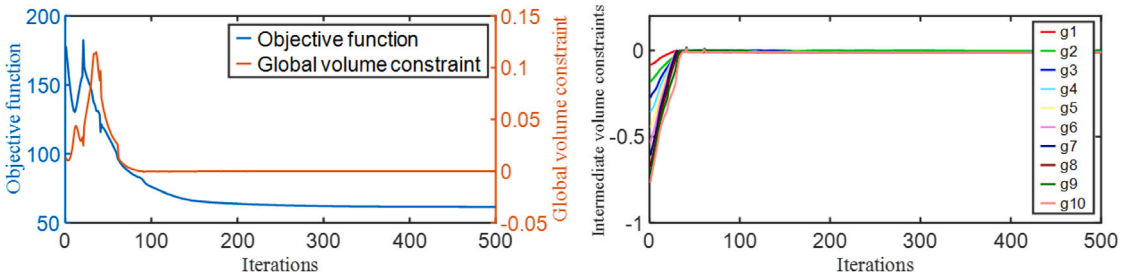


Fig. 13. The convergence curves of an optimized 3D structure corresponding to $\alpha_i = 0.01$. Left: the objective function and global volume constraint. Right: the volume constraints for all intermediate structures.

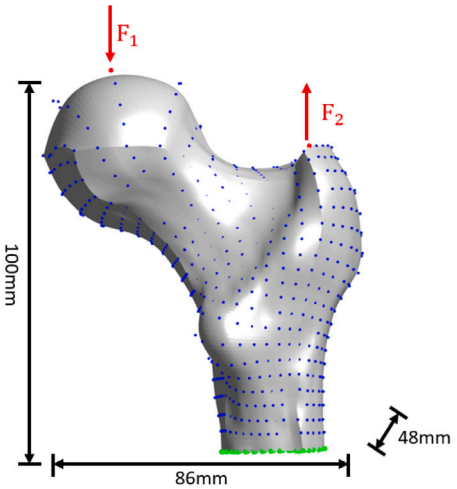


Fig. 14. The visualization of the 3D Bone design domain, control points (the blue points), dimensions, heat sources (the green control points), and boundary (the green control points) / loading (the red control points) conditions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The optimized virtual time fields, components, and sequences are shown in Fig. 15, where the last two rows show the optimized sequences in the front and back viewpoints, respectively. The manufacturing sequence is optimized to consist of 20 layers. From this figure we can observe that, due to the self-weight, the top and middle holes enlarge, while the bottom hole disappears, and more solid material moves to the lower regions. The layers of the sequence optimized using $\alpha_i = 0.05$ are shown in Fig. 16. To visualize the movement of solid materials, the design domain is uniformly divided into 10 subregions along the z-axis. The number of Gauss points with solid material is visualized in Fig. 17(a). We can clearly observe that as the value of α_i increases, more solid material shifts from the top subregions to the bottom subregions to counteract the effects of self-weight. The compliances of the entire components and the intermediate structures are shown in Fig. 17(b) and (c), respectively. As we can see, as the value of α_i increases, the compliance of the entire component also increases. When α_i changes from 0.00 to 0.15, the compliance of the entire component changes from 19.41 to 27.88, resulting in an increase of 43.64%. In contrast, the compliance of the intermediate structure experiences a substantial reduction as the value of α_i increases. For instance, the compliance of the eighteenth intermediate structure changes from 80.31 when $\alpha_i = 0.00$ to 18.27 when $\alpha_i = 0.15$, achieving a reduction of 77.25%. Compared to the increase in the compliance of the entire component, the reduction in the compliances of the intermediate structures is much more significant. The convergence curves of the objective functions for all values of α_i are shown in Fig. 17(d). These curves demonstrate that the objective functions have nearly converged after 200 iterations, with very similar values.

4.3. Layer thickness control

The manufacturing sequences optimized in the previous sections satisfy the volume constraints of the intermediate structures and reduce the compliance caused by the self-weight of these structures. However, the significant variation in the thickness of each layer makes it challenging to fabricate using AM technologies. To improve the manufacturability of the generated sequence, layer thickness should be considered in the optimization process.

As presented in [46], for a scalar field f, if $\|\nabla f\| \equiv a$, where a is a constant and ∇ is the gradient operator, its isolines (2D) or isosurfaces (3D) will be equidistant. Building on this theory, several articles have addressed layer thickness to generate curved

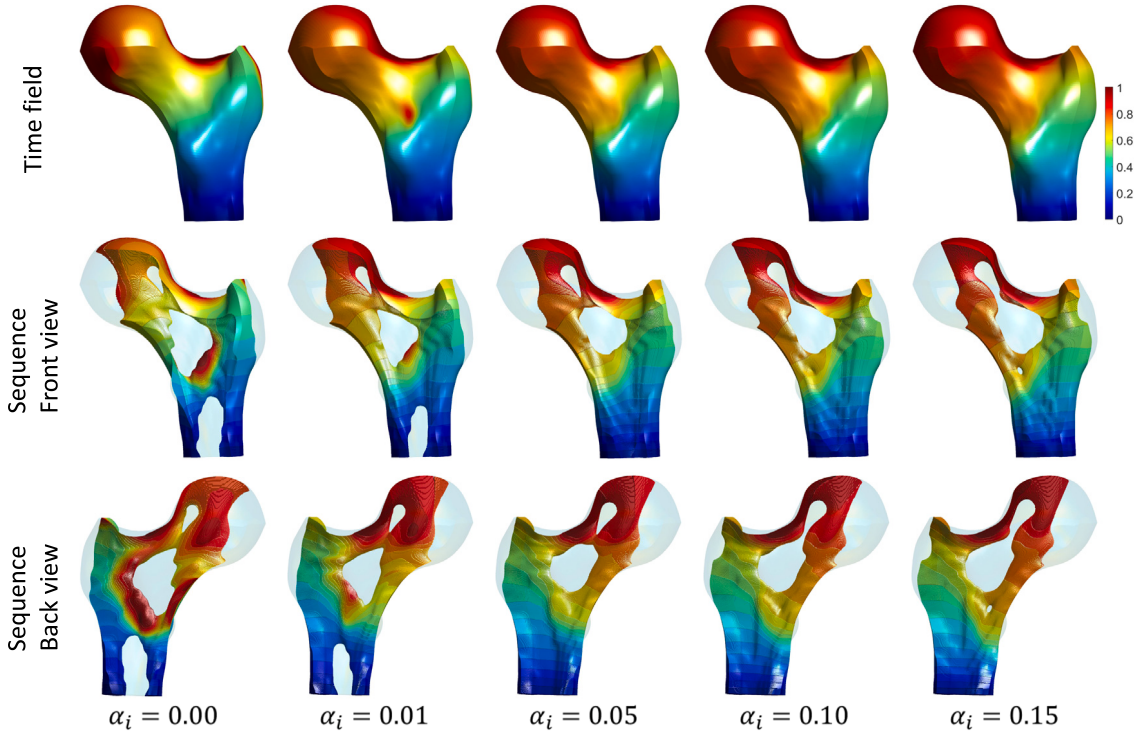


Fig. 15. Optimized virtual time fields (the first row) and sequences using five different values of α_i: 0.00, 0.01, 0.05, 0.10, and 0.15 from left to right. The last two rows show the sequences from two different viewpoints.

layers of uniform thickness for multi-axis 3D printing [47,48]. However, their methods cannot be directly applied to the proposed optimization formulation, as the upper bound of the virtual time field is 1. Consequently, it becomes extremely challenging to select an appropriate value for a to generate a manufacturing sequence with uniform layer thickness while achieving the desired number of layers and satisfying the volume constraint of the intermediate structures. Instead, a is treated as a design variable in our formulation. Based on the VTF defined in Eq. (18) and the theory provided in [46], a constraint for achieving uniform layer thickness for 2D case is proposed as follows:

$$g_l = \int_\Omega \left(\|\nabla \mathcal{T}(\boldsymbol{\mu}, \xi, \eta)\|_2 - a \right)^2 d\Omega \leq \epsilon, \tag{38}$$

where $\|.\|_2$ represents the l_2-norm of a vector, ϵ is a constant used to control the variation of the layer thickness which is set to 10^{-4} in this work, as 3D printing technologies permit small changes in thickness within a layer.

The derivative of the layer thickness constraint g_l with respect to the virtual heat conductive coefficient $\mu_{i,j}$ is calculated as:

$$\begin{aligned} \frac{\partial g_l}{\partial \mu_{i,j}} &= \int_\Omega 2 \sum_{\tilde{i}=1}^{n} \sum_{\tilde{j}=1}^{m} \frac{\left(\nabla R^{p,q}_{\tilde{i},\tilde{j}}(\boldsymbol{\mu}, \xi, \eta) \right)^{\mathsf{T}} \nabla \mathcal{T}(\boldsymbol{\mu}, \xi, \eta)}{\|\nabla \mathcal{T}(\boldsymbol{\mu}, \xi, \eta)\|_2} \frac{\partial t_{\tilde{i},\tilde{j}}}{\partial \mu_{ij}} \left(\|\nabla \mathcal{T}(\boldsymbol{\mu}, \xi, \eta)\|_2 - a \right) d\Omega \\ &= -\int_\Omega 2 \sum_{\tilde{i}=1}^{n} \sum_{\tilde{j}=1}^{m} \frac{\left(\nabla R^{p,q}_{\tilde{i},\tilde{j}}(\boldsymbol{\mu}, \xi, \eta) \right)^{\mathsf{T}} \nabla \mathcal{T}(\boldsymbol{\mu}, \xi, \eta)}{\|\nabla \mathcal{T}(\boldsymbol{\mu}, \xi, \eta)\|_2} \frac{\partial T_{\tilde{i},\tilde{j}}}{\partial \mu_{ij}} \left(\|\nabla \mathcal{T}(\boldsymbol{\mu}, \xi, \eta)\|_2 - a \right) d\Omega, \end{aligned} \tag{39}$$

where $\frac{\partial T_{i,j}}{\partial \mu_{ij}}$ can be calculated using Eq. (10). The derivative of g_l with respect to a is calculated as:

$$\frac{\partial g_l}{\partial a} = -\int_\Omega 2 \left(\|\nabla \mathcal{T}(\boldsymbol{\mu}, \xi, \eta)\|_2 - a \right) d\Omega. \tag{40}$$

The integrals defined in the above equations can be solved using the Gauss quadrature method.

The layer thickness constraint defined in Eq. (38) is incorporated into the optimization formulation (24) and evaluated over the design domain shown in Fig. 4. Apart from the layer thickness constraint, all other settings are identical to those in the experiment conducted in Section 4.1. The generated time field and manufacturing sequence are shown in Fig. 18. The weighting factor α_i is set to 0.5 for the objective function. Fig. 18(a) and (b) show the optimized time field and sequence without considering the layer thickness constraint, while Fig. 18(c) and (d) show the optimized time field and sequence with this constraint. It is evident that the

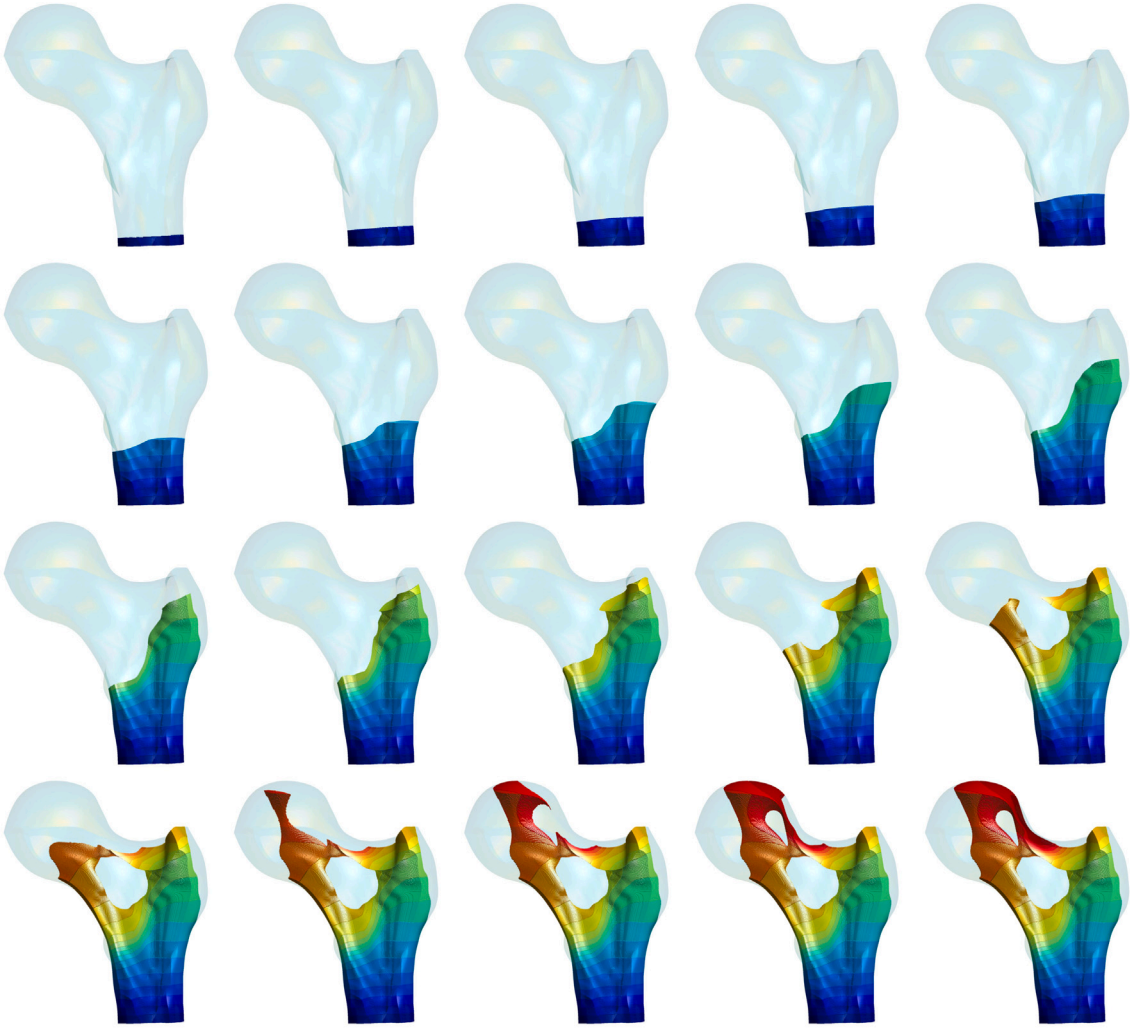


Fig. 16. The intermediate structures for the optimized sequence using $\alpha_i = 0.05$.

sequence in Fig. 18(d) exhibits a much more uniform layer thickness compared to that in Fig. 18(b). The convergence curves of the objective function for the results shown in Fig. 18 are plotted on the left-hand side of Fig. 19. Both optimizations demonstrate good convergence. The objective function of the sequence generated without the layer thickness constraint is slightly lower than that of the sequence with the constraint. The compliances of the intermediate structures for both sequences are shown on the right-hand side of Fig. 19. Similar to the objective function, the compliances of the intermediate structures in the sequence generated without the layer thickness constraint are lower than those in the sequence generated with the constraint. The increase in both the objective function and the compliances of the intermediate structures in the latter sequence is due to the incorporation of the additional layer thickness constraint. The convergence curves of the layer thickness constraint and the volume constraints for the intermediate structures are shown in Fig. 20. As we can see that all constraints are nearly satisfied after approximately 100 iterations, and they are fully met thereafter. The layers of the sequence shown in Fig. 18(d) are visualized in Fig. 21.

We also test our method on an irregular design domain, that is a Quarter Annulus shown in Fig. 22(a). The control points on the left edge are fixed (the green points). The control point at the bottom-right corner is subjected to a force of 1 kN directed towards the left (the red arrow). The control points on the bottom edge serve as heat sources (the blue control points). The weighting factor α_i in the objective function is set to 0.5. The optimized results are shown in Fig. 22. The time field and sequence generated without the layer thickness constraint are shown in Fig. 22(b) and (c), resulting in a significant variation in layer thickness. When the layer thickness constraint is applied, the generated sequence exhibits a much more uniform layer thickness, see the layers in Fig. 22(e). The convergence curves of the objective function and the compliances of the intermediate structures for the generated sequences using different methods are shown in Fig. 23. From this figure we can see that, due to the addition of an extra constraint, both the objective function value and the compliances of the intermediate structures are slightly higher than those generated without considering this constraint. However, the objective functions for both methods stabilize after about 200 iterations. The convergence

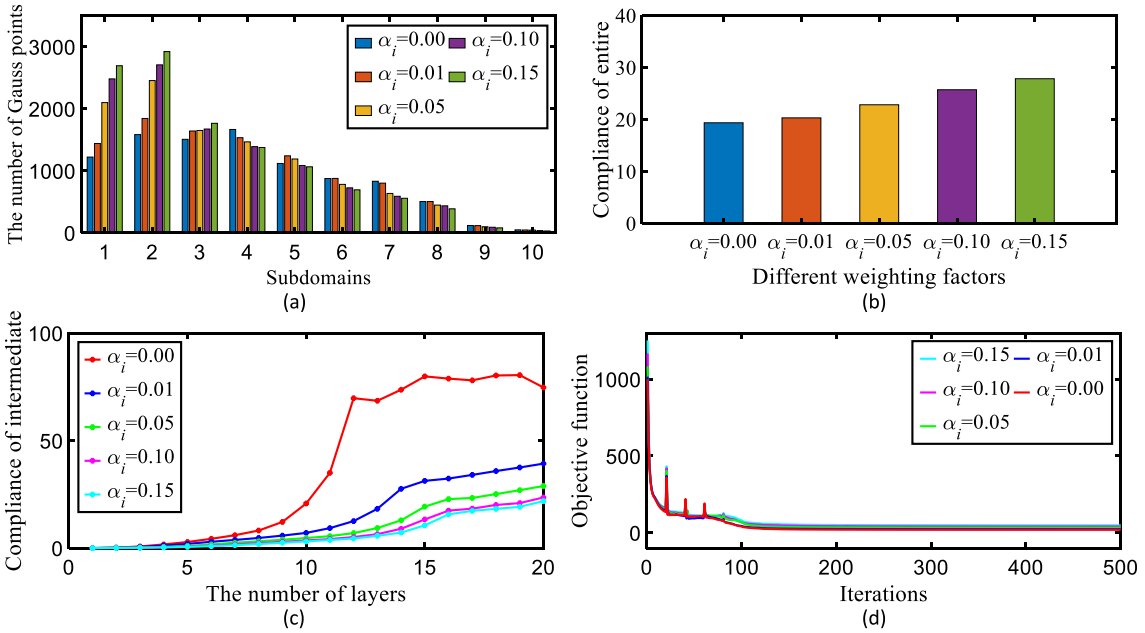


Fig. 17. Visual representations of different results. The variations in the solid Gauss points along the z-axis direction are visualized in (a). The compliances of the components and intermediate structures are displayed in (b) and (c), respectively. The convergence curves of the objective function are plotted in (d).

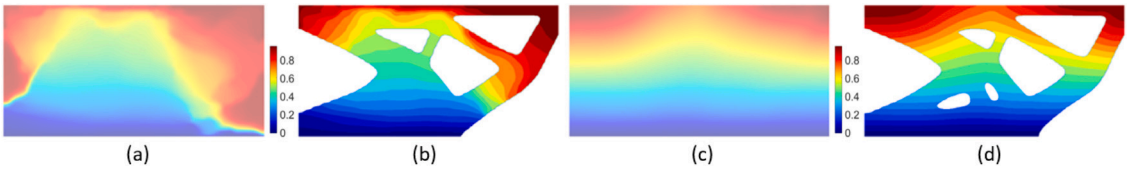


Fig. 18. The experimental results of a cantilever beam at $\alpha_i = 0.5$: (a) and (b) show the optimized virtual time field and sequence without the layer thickness constraint, while (c) and (d) the optimized time field and sequence with this constraint.

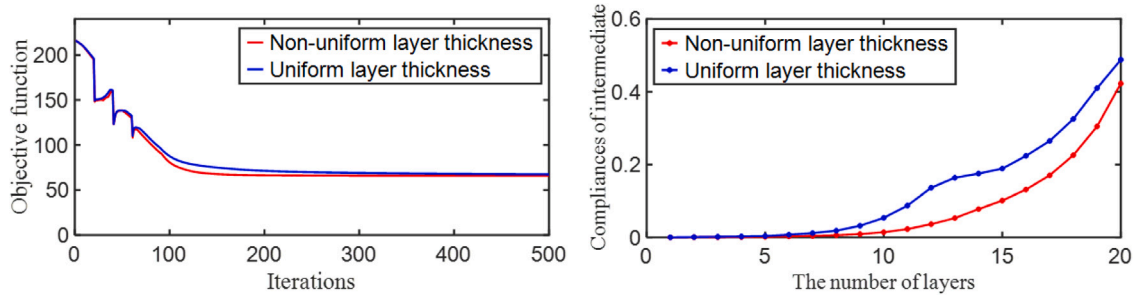


Fig. 19. Left: The convergence curves of the objective function for the results shown in Fig. 18. Right: The compliances of the intermediate structures for both sequences.

curves of the layer thickness constraint and volume constraints for the intermediate structures are plotted in Fig. 24. This figure clearly demonstrates that all constraints are effectively satisfied.

As shown in Fig. 25, the proposed method is tested on the L-Shaped design domain, which is a well-known benchmark problem in TO. The boundary condition, loading condition, and the dimensions are shown in Fig. 25(a). The control points on the top edge are fixed, and the control point at the top-right corner has a downward force of 1 kN. The control points on the bottom edge act as heat sources. The weighting factor α_i in Eq. (24) is set to 0.5. The maximum available volume fraction is 0.4. The virtual time field and manufacturing sequence shown in Fig. 25(b) and (c) are optimized without considering the layer thickness constraint. We can observe that the gradient of the virtual time field is irregular, and the thickness of several layers is highly non-uniform. As contrast, the virtual time field and manufacturing sequence shown in Fig. 25(d) and (e) are optimized with the layer thickness constraint.

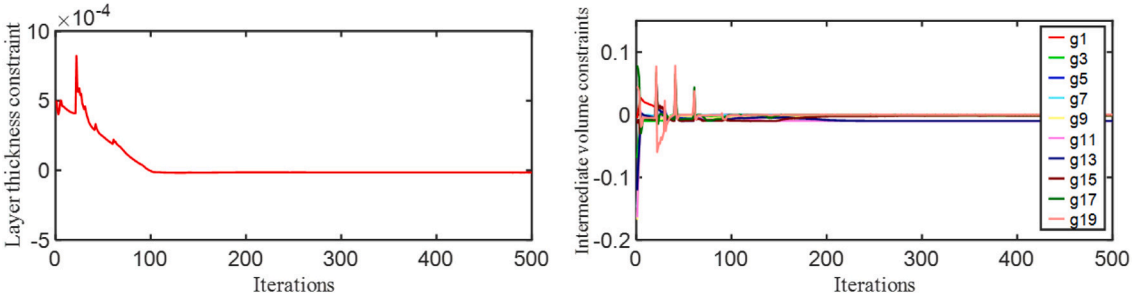


Fig. 20. Left: the convergence curves of the layer thickness constraint. Right: the volume constraints of the intermediate structures for the results shown on the right-hand side of Fig. 18.

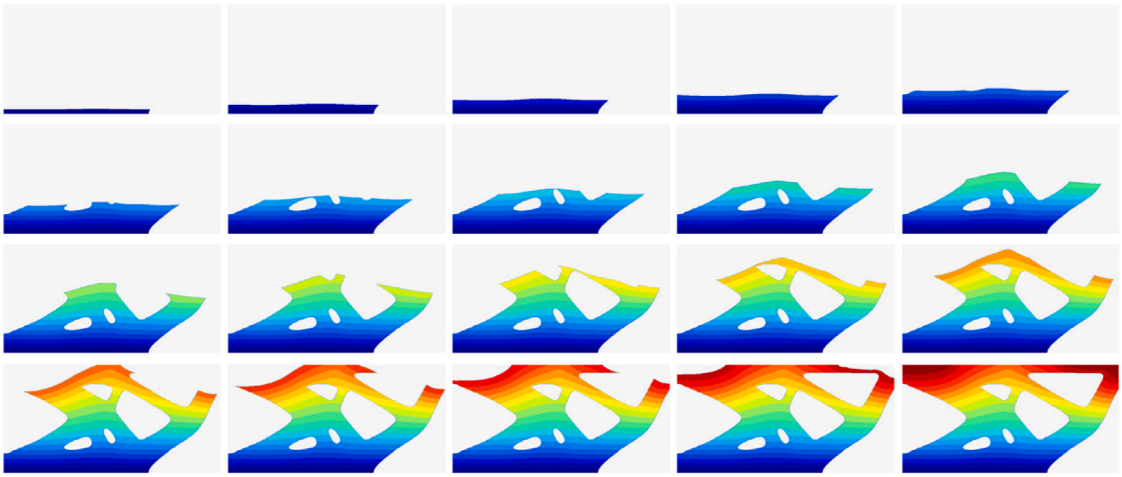


Fig. 21. Visualization of the intermediate structures of the manufacturing sequence shown in Fig. 18(d).

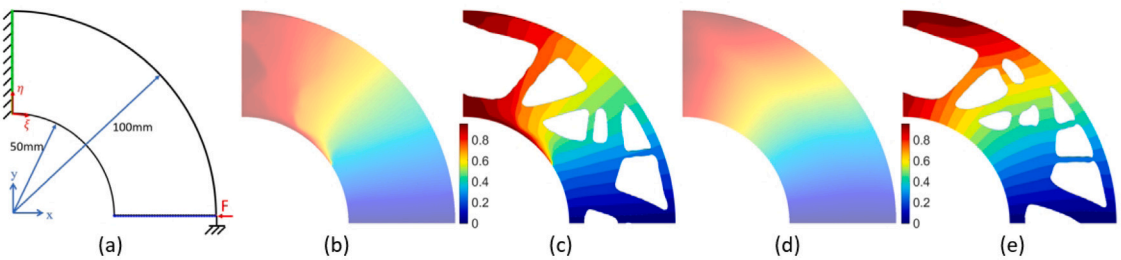


Fig. 22. Experimental results for a Quarter Annulus (a). (b) and (c) show the optimized virtual time field and sequence without the layer thickness constraint, whereas (d) and (e) are generated with this constraint. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

It is clear that the uniformity of the virtual time field and layer thickness have been significantly improved. The compliance of the optimized components shown in Fig. 25(c) and (e) are 239.86 and 274.84, respectively. The experiments conducted in this subsection demonstrate the robustness and effectiveness of the proposed layer thickness constraint for both regular and irregular 2D design domains.

To further validate the effectiveness of the proposed layer thickness constraint, a 3D experiment is conducted on the design domain shown in Fig. 26(a). Apart from the loading conditions and the maximum available volume fraction, all other settings remain identical to the experiment detailed in Section 4.2. As depicted in Fig. 26(a), the control points located at the top-right corners experience downward forces of 1 kN. The maximum available volume fraction is set to 0.4. The virtual time field and manufacturing sequence presented in Fig. 26(b) and (c) are optimized without the layer thickness constraint. As a result, the gradient of the virtual time field is irregular, and the resulting sequence exhibits significant non-uniformity in layer thickness. In contrast, the virtual time field and manufacturing sequence shown in Fig. 26(d) and (e) are generated with the layer thickness constraint.

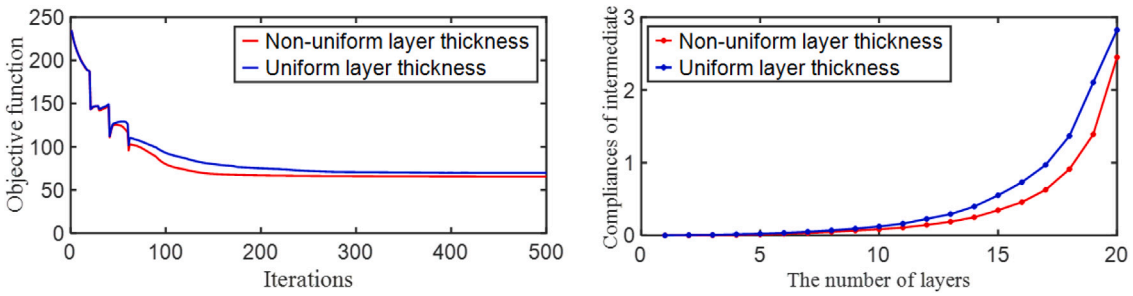


Fig. 23. Left: the convergence curves of the objective function for the results shown in Fig. 22. Right: the compliances of the intermediate structures for both sequences.

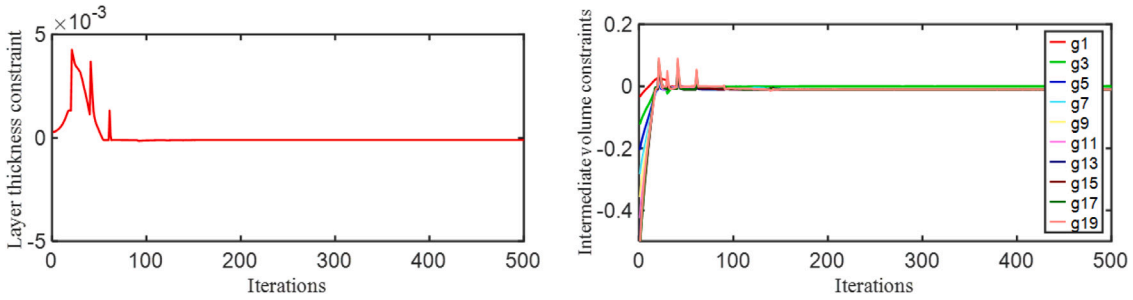


Fig. 24. Left: the convergence curves of the layer thickness constraint. Right: the volume constraints of the intermediate structures for the results shown in Fig. 22(e).

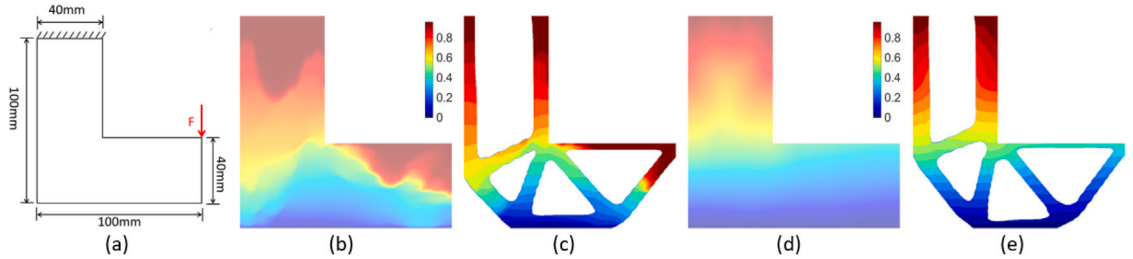


Fig. 25. Experimental results for the L-Shaped design domain (a). The optimized virtual time field and sequence without the layer thickness constraint are shown in (b) and (c), respectively, whereas (d) and (e) are generated with this constraint.

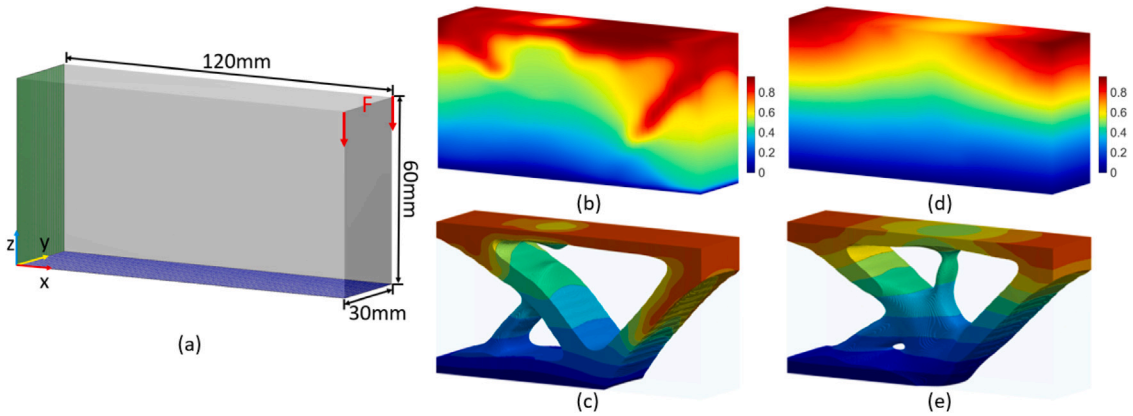


Fig. 26. Experimental results for the 3D Cantilever beam (a). The optimized virtual time field and sequence without the layer thickness constraint are presented in (b) and (c), respectively, while (d) and (e) show the results generated with this constraint.

It is clear that both the gradient of the virtual time field and the layer thickness of the sequence are significantly improved. The compliance of the optimized components shown in Fig. 26(c) and (e) are 144.73 and 151.21, respectively.

The experiments conducted in this subsection demonstrate that the proposed layer thickness constraint performs well in both 2D and 3D cases, with only a slight sacrifice in the performance of the obtained component. For the experiment, the design domain, as well as the boundary and loading conditions, are the same as those shown in Fig. 4.

5. Discussions

In this section, we will discuss the efficiency of the proposed algorithm in Section 5.1, the influence of the degree of the basis functions in Section 5.2, and the influence of the number of Gauss quadrature points in Section 5.3.

5.1. Algorithm efficiency

In each iteration of the optimization, the static equilibrium equation (5), the heat conduction equation (10), and a set of linear systems for calculating the compliances of the intermediate structures and performing the sensitivity analysis must be solved. This significantly impacts the efficiency of the proposed method. The running time for the experiments conducted in the previous section is reported in Table 1. The time required to solve the static equilibrium equation (5) and the heat conduction equation (10) in a single iteration is listed in the second and third columns of Table 1, respectively. For each linear system, the scale is consistent across all results within a single figure, resulting in the same solving time. Therefore, for each figure, we report the solving time for each equation only once. Since the scale of the static equilibrium equation is twice that of the heat conduction equation, and the assembly of the stiffness matrix for the static equilibrium equation is considerably more complex, it requires more time to solve. For instance, in each iteration, solving Eq. (5) takes approximately 2.4907 s, while Eq. (10) takes 0.1511 s for the 3D Bone. It should be noted that, whether or not the layer thickness constraint is applied, the scale of the equations remains the same. Consequently, the time required to solve each equation is identical for the same design domain. The total running time for all examples is listed in the last column of Table 1. We observe that the 2D cases take approximately one hour, while the 3D cases require about 10 h. For instance, the 2D-CB requires approximately 3.5×10^3 s on average, while the 3D-CB takes over 3×10^4 s on average. Additionally, the running time for $\alpha_i = 0$ is clearly shorter than those for other values of α_i. For instance, for 3D Bone, the running time when $\alpha_i = 0$ is 9483 s, while for the other values of α_i, the average running time exceeds 2.4×10^4 s. The main reason is that the compliances of the intermediate structures are not calculated when $\alpha_i = 0$. As a result, several large-scale linear systems do not need to be solved for the objective function and sensitivity analysis. In practical engineering applications, solving 3D problems is necessary, so the efficiency of the proposed algorithm should be improved to enhance its applicability.

5.2. Influence of the degree of the basis functions

The values of p and q used in Eq. (1) determine the degree of the basis functions and the number of Gauss quadrature points. As a result, they may affect the accuracy of the IGA. In this subsection, an experiment is conducted on the design domain shown in Fig. 4 to investigate the influence of the (p, q) pair on the performance of the proposed method. When the $(p, q) = (1, 1)$, the resulting basis functions are linear, which introduce significant errors in the computation. When the values of p and q exceed 4, the efficiency of the algorithm becomes too low. Therefore, their lower and upper bounds are set to 2 and 4, respectively, in this experiment. Three sets of (p, q) pairs are evaluated, which are $(2, 2)$, $(3, 3)$, and $(4, 4)$. According to the rule mentioned in Section 2.2, the corresponding the numbers of Gauss quadrature points are 3×3, 4×4, and 5×5, respectively. The testing results using these (p, q) pairs are shown in Fig. 27, where the top row shows the optimized virtual time fields and the bottom row shows the optimized manufacturing sequences. From left to right, the corresponding (p, q) pairs are $(2, 2)$, $(3, 3)$, $(4, 4)$, respectively. From this figure we can clearly see that the virtual time fields and manufacturing sequences generated with different (p, q) pairs are similar. The total running time and the compliances of these components are listed in Table 2. Theoretically, higher-degree basis functions provide smoother and potentially more accurate solutions. However, the components generated with different (p, q) pairs exhibit similar compliances, with the maximum difference between them being less than 5%. More importantly, as the values of p and q increase, the total running time of the proposed algorithm increases rapidly. Therefore, (p, q) pair for the 2D Cantilever Beam is set to $(2, 2)$ in this experiment to balance computational accuracy and efficiency. The degrees of the basis functions for the other design domains are provided in Appendix C.

5.3. Influence of the number of Gauss quadrature points

In this subsection, an experiment is conducted to investigate the influence of the number of Gauss quadrature points on the performance of the proposed method when $(p, q) = (2, 2)$. Four different numbers of Gauss quadrature points are evaluated: 3×3, 4×4, 5×5, and 6×6. The optimized virtual time fields and manufacturing sequences are shown in Fig. 28, with the numbers of Gauss quadrature points being 3×3, 4×4, 5×5, and 6×6 from left to right. We can observe from this figure that the differences between these virtual time fields and sequences are difficult to discern. The total running time and the compliances of the generated components under different numbers of Gauss quadrature points are listed in Table 3. It can be seen that all components have almost the same compliance. However, as the number of Gauss quadrature points increases, the total running time increase rapidly. From this experiment, we conclude that using a sufficient number of Gauss quadrature points results in minimal variation in numerical results, but significantly impacts computation time. Therefore, the rule mentioned in Section 2.2 is applied to determine the number of Gauss quadrature points based on the (p, q) pair.

Table 1
Statistics on the efficiency of the proposed method. The second and third columns list the time required to solve Eq. (5) and Eq. (10) in a single iteration, and the last column reports the total running time. Fig$i$-jc represents the jth column of Figure i. "2D-CB", "2D-QA", "2D-LS", and "3D-CB" represent the 2D Cantilever Beam, 2D Quarter Annulus, 2D L-Shaped, and 3D Cantilever Beam, respectively. "T" represents the layer thickness constraint is considered.

Model	Solving Eq. (5) (s)	Solving Eq. (10) (s)	Total time (s)
2D-CB	0.3062	0.2091	Fig. 5–1c: 822 Fig. 5–2c: 2793 Fig. 5–3c: 3414 Fig. 5–4c: 3575 Fig. 5–5c: 3571
2D-CB-T	0.3062	0.2091	Fig. 18(d): 2804
2D-QA	0.6938	0.5455	Fig. 22(c): 6065
2D-QA-T	0.6938	0.5455	Fig. 22(e): 6270
2D-LS	0.3072	0.2128	Fig. 25(c): 3326
2D-LS-T	0.3072	0.2128	Fig. 25(e): 3441
3D-CB	2.6362	0.2707	Fig. 10–1c: 4059 Fig. 10–2c: 33 181 Fig. 10–3c: 31 954 Fig. 10–4c: 34 839 Fig. 10–5c: 32 224
3D-CB-T	2.6362	0.2707	Fig. 26(c): 35 078 Fig. 26(e): 36 108
3D Bone	2.4907	0.1511	Fig. 15–1c: 9483 Fig. 15–2c: 24 919 Fig. 15–3c: 28 758 Fig. 15–4c: 33 483 Fig. 15–5c: 33 956

Table 2
The total running time (third column) of the proposed algorithm using different (p, q) pairs (first column). The second column shows the number of Gauss quadrature points, and the last column displays the compliances of the generated components.

(p, q) pairs	Number of gauss points	Total running time (s)	Compliance
2×2	3×3	2804	67.55
3×3	4×4	5051	69.27
4×4	5×5	38 763	69.98

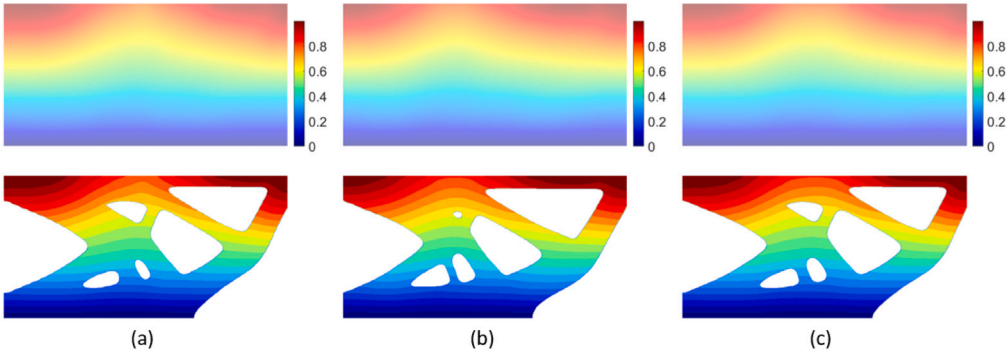


Fig. 27. The virtual time fields (top row) and the manufacturing sequences (bottom row) generated with three different (p, q) pairs, which are $(2, 2)$, $(3, 3)$, and $(4, 4)$ from left to right.

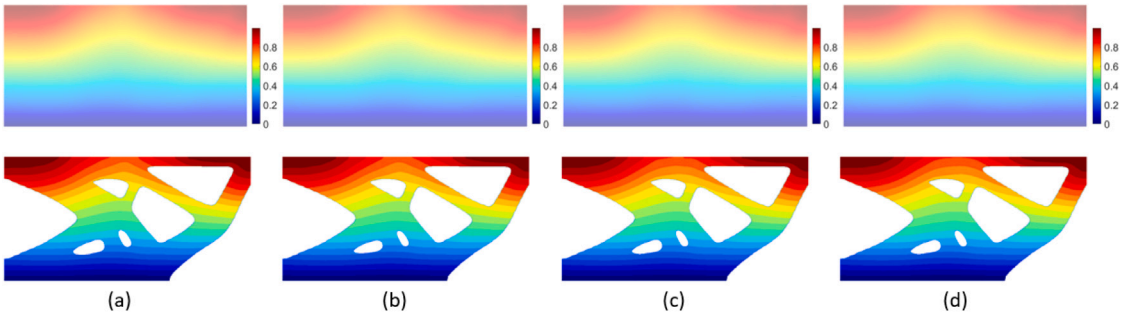


Fig. 28. The virtual time fields (top row) and the manufacturing sequences (bottom row) generated with four different numbers of Gauss quadrature points when $(p, q) = (2, 2)$, namely 3×3, 4×4, 5×5, and 6×6 from left to right.

Table 3
The total running time (second column) of the proposed algorithm using different numbers of Gauss quadrature points (first column) when $(p, q) = (2, 2)$. The last column shows the compliances of the generated components.

Number of gauss points	Total running time (s)	Compliance
3×3	2804	67.5537
4×4	3442	67.5087
5×5	4232	67.5058
6×6	5265	67.5054

5.4. Comparisons

In this subsection, we first conduct an experiment by replacing the VTCCF defined in Eq. (16) with the following equation to solve the heat conduction equation in the optimization:

$$\tilde{\mathcal{M}}(\boldsymbol{\mu}, \boldsymbol{\rho}, \xi, \eta) = \mathcal{M}(\boldsymbol{\mu}, \xi, \eta) * \mathcal{X}(\boldsymbol{\rho}, \xi, \eta). \tag{41}$$

In this experiment, all settings are the same as those used in Section 4.3, while the weighting factors α_is in the formulation (24) are set to 0. The generated virtual time field and manufacturing sequence are shown on the left hand side of Fig. 29. As a comparison, the virtual time field and manufacturing sequence shown on the right hand side of Fig. 29 are generated using the method proposed in this work, which uses the VTCCF defined in Eq. (16) to solve the heat conduction equation. The zoomed-in regions are used to visualize the layer thickness. From this figure we can see that the generated components have almost the same topology and they have similar compliances which are 66.78 and 65.52, respectively. However, the layer thicknesses of the manufacturing sequence generated using Eq. (41) are not as uniform as those of the sequence generated using Eq. (16). The main reason is that coupling the VTCCF and DDF to solve the heat conduction equation makes it more challenging to control the gradient of the generated temperature field, especially at the boundaries between the solid and void regions. Additionally, coupling the VTCCF and DDF increases the complexity of solving the heat conduction equation, which can lead to lower computational efficiency. The running time for the results on the left-hand side of Fig. 29 is 2246 s, while it is 1165 s for the results on the right-hand side.

An experiment is conducted to demonstrate the advantages of the ITO method over the FEM-based TO method for generating manufacturing sequences. Specifically, the proposed method is compared with the state-of-the-art method presented in [40]. Since

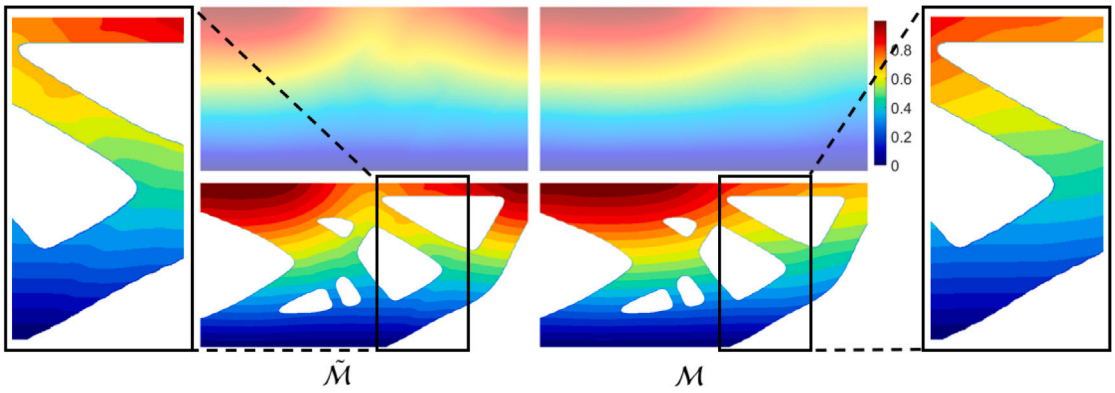


Fig. 29. The virtual time field and manufacturing sequence shown on the left hand side are obtained by solving heat conduction equation with the Eq. (41), while the results on the right hand side are generated using the Eq. (16). The zoomed-in regions are used to visualize the layer thickness.

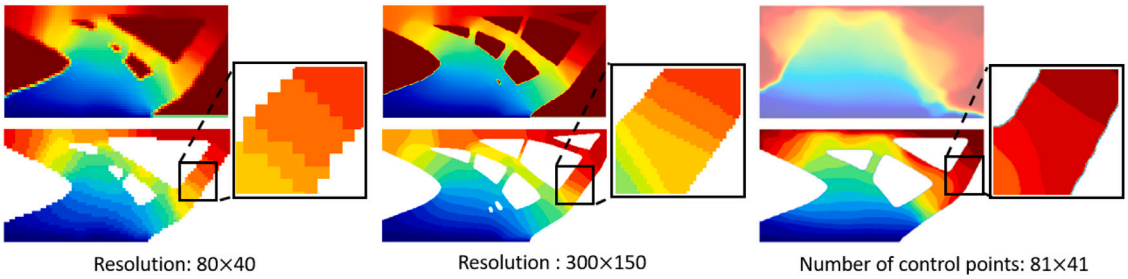


Fig. 30. The virtual time field and manufacturing sequence generated using the method presented in [40] with a resolution of 80 × 40 (left) and a resolution of 300 × 150 (middle), with the results in the right generated using the proposed method with 81 × 41 control points. The zoomed-in regions visualize the smoothness of the boundaries of the components and layers.

the layer thickness is not considered in [40], it is excluded from this comparison. The weighting factors α_is in the formulation (24) are set to 0.5. For the FEM-based method [40], the design domain is discretized into hexahedral finite element meshes with two different resolutions: 80 × 40 and 300 × 150. The number of layers is set to 20. The generated results are shown in Fig. 30. The virtual time fields and manufacturing sequences in the left and middle columns are generated using the method proposed in [40], while the results shown in the right column are generated using our method. The resolutions of the finite element meshes for the FEM-based method and the number of control points for the proposed method are listed at the bottom. These components have similar compliances which are 62.18, 63.59, and 65.87, respectively. We can observe from this figure that, although the method proposed in [40] is significantly more efficient than our method when the finite element mesh resolution is 80 × 40 (taking 966 s and 2793 s, respectively), both the boundaries of the component and the layers of the manufacturing sequence exhibit a severe staircase effect. When the resolution is increased to 300 × 150, the smoothness of the boundaries is significantly improved, as seen in the zoomed-in region of the middle image. However, the boundaries of the results in the middle image remain less smooth compared to those in the right image which is generated using only 81 × 41 control points. More importantly, generating the results in the middle image requires significantly more time compared to those in the right image, with times of 10 932 s and 2793 s, respectively. From this comparison, we can clearly see the advantage of the proposed IGA-based space–time TO formulation for generating the manufacturing sequence, compared to the FEM-based space–time TO [40], which is the state-of-the-art method for manufacturing sequence generation.

6. Conclusions

In this paper, we present a space–time isogeometric TO formulation to concurrently optimize the structure shape and the additive manufacturing sequence. In the proposed method, a density distribution function (DDF) is constructed by combining NURBS basis functions with the density values at the control points to represent the material distribution. Similarly, a PTF is introduced, following the same approach as the DDF, to represent the virtual build time of the material. The combination of the DDF and the PTF enables the generation of the manufacturing sequence. To prevent the generation of isolated parts that cannot be fabricated using additive manufacturing technologies, the heat conduction equation is employed to generate a virtual temperature field at the control points, which is then transformed into a pseudo-time field. This approach ensures that the generated pseudo-time fields are free of local minima. Our method defines both the density field and the virtual heat conduction coefficients at the control points, which are

treated as design variables to be optimized. To ensure the manufacturability of the generated manufacturing sequence, a layer thickness constraint has been proposed for the 2D case, based on the gradient of the PTF. The effectiveness of the proposed optimization formulation has been demonstrated by considering the self-weight for both 2D and 3D cases.

Physical validation of the generated manufacturing sequence will be a focus of our future research. To achieve this, considerations for collision-free and self-supporting structures must be integrated into the generation of the manufacturing sequence [47,48]. For all intermediate structures, the weighting factors $\{\alpha_i\}$ in the objective function of formulation (24) is set to the same value. Depending on the demands in different applications, $\{\alpha_i\}$ can be further optimized to emphasize the importance of specific layers. Since each iteration requires solving the static equilibrium equation, the heat conduction equation, and a set of other large-scale linear systems, the efficiency of the proposed method is significantly impacted, particularly in 3D cases. In our future work, we plan to accelerate the process through parallel computing such as by a GPU-based implementation.

CRediT authorship contribution statement

Li Yang: Writing – original draft, Visualization, Validation, Methodology, Formal analysis. **Weiming Wang:** Writing – review & editing, Visualization, Validation, Methodology, Formal analysis, Conceptualization. **Ye Ji:** Visualization, Software, Resources, Formal analysis, Data curation. **Chun-Gang Zhu:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization. **Charlie C.L. Wang:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declared that they have no conflict of interest exists in the submission of this manuscript, and manuscript is approved by all authors for publication.

Acknowledgments

The authors wish to thank the National Natural Science Foundation of China (No. 62172073, 12071057, 12471358), and the Science and Technology Project of Chongqing Education Committee (Grant No. KJZD-K202400514). Weiming Wang and Li Yang are partially supported by the UK Engineering and Physical Sciences Research Council (EPSRC) Grant (EP/W024985/1). The authors would like to express their gratitude to Depeng Gao from Zhejiang University for kindly providing the Femur model.

Appendix A. Details of nurbs basis function

The bivariate NURBS basis function is defined as follows:

$$R_{i,j}^{p,q}(\xi,\eta)=\frac{N_{i,p}(\xi)M_{j,q}(\eta)\omega_{ij}}{\sum_{\hat{i}=1}^{n}\sum_{\hat{j}=1}^{m}N_{\hat{i},p}(\xi)M_{\hat{j},q}(\eta)\omega_{\hat{i}\hat{j}}}, \tag{A.1}$$

where $N_{i,p}$ and $M_{j,q}$ are the univariate B-spline basis functions in two parametric directions, respectively, and $\omega_{ij}\geq 0$ indicates the weight associated to the control point $\mathbf{P}_{i,j}$. The B-spline basis function is defined recursively using the Cox-de-Boor formula [49], starting with the piecewise constant ($p=0$),

$$N_{i,0}(\xi)=\begin{cases}1, & \text{if } \xi_i\leq\xi\leq\xi_{i+1},\\ 0, & \text{otherwise.}\end{cases} \tag{A.2}$$

If $p\geq 1$, the corresponding basis function is defined as:

$$N_{i,p}(\xi)=\frac{\xi-\xi_i}{\xi_{i+p}-\xi_i}N_{i,p-1}(\xi)+\frac{\xi_{i+p+1}-\xi}{\xi_{i+p+1}-\xi_{i+1}}N_{i+1,p-1}(\xi), \tag{A.3}$$

where ξ_i represents a knot in a parametric direction. $M_{j,q}$ can be defined in a similar way. It is worth noticing that fractions with the form $\frac{0}{0}$ are defined as zero in Eq. (A.3).

Appendix B. Definition of ψ

The Shepard function ψ is defined as follows:

$$\psi(\rho_{i,j})=\varphi(\rho_{i,j})/\sum_{\hat{i}=1}^{\bar{n}}\sum_{\hat{j}=1}^{\bar{m}}\varphi(\rho_{\hat{i},\hat{j}}), \tag{B.1}$$

where the function $\varphi(.)$ is defined using compactly supported radial basis functions with C^4 continuity [50]:

$$\varphi(x)=(1-x)_{+}^{6}(35x^2+18x+3). \tag{B.2}$$

Table C.4
Parameterization details of the regular design domains tested in this work. "2D-CB", "2D-QA", "2D-LS", and "3D-CB" represent the 2D Cantilever Beam, 2D Quarter Annulus, 2D L-Shaped, and 3D Cantilever Beam, respectively. "DE" and "KN" represent the degree of elevation and the number knots, respectively. The second to fourth columns represent the control points, weights, and knot vector information of the initial parameterization. The "Degree" column specifies the actual degree of the NURBS basis functions, the "CP-Num" indicates the corresponding number of control points utilized, and the "Nel" represents the number of IGA elements.

Model	Control points(x, y, z)	Weights	Knots vector	DE	KN	Degree	CP-Num	Nel
2D-CB	$\mathbf{P}_{1,1} = (0, 0)$, $\mathbf{P}_{2,1} = (1.2, 0)$, $\mathbf{P}_{1,2} = (0, 0.6)$, $\mathbf{P}_{2,2} = (1.2, 0.6)$.	$\omega_{1,1} = 1$, $\omega_{2,1} = 1$, $\omega_{1,2} = 1$, $\omega_{2,2} = 1$.	$\xi = [0, 0, 1, 1]$, $\eta = [0, 0, 1, 1]$.	[1, 1]	80 × 40	[2, 2]	81 × 41	79 × 39
2D-QA	$\mathbf{P}_{1,1} = (0, 0.5)$, $\mathbf{P}_{2,1} = (0.5, 0.5)$, $\mathbf{P}_{3,1} = (0.5, 0)$, $\mathbf{P}_{1,2} = (0, 1)$, $\mathbf{P}_{2,2} = (1, 1)$, $\mathbf{P}_{3,2} = (1, 0)$.	$\omega_{1,1} = 1$, $\omega_{2,1} = \sqrt{2}/2$, $\omega_{3,1} = 1$, $\omega_{1,2} = 1$, $\omega_{2,2} = \sqrt{2}/2$, $\omega_{3,2} = 1$.	$\xi = [0, 0, 0, 1, 1, 1]$, $\eta = [0, 0, 1, 1]$.	[1,2]	80 × 40	[3, 3]	82 × 42	79 × 39
2D-LS	$\mathbf{P}_{1,1} = (0, 1)$, $\mathbf{P}_{2,1} = (0, 0)$, $\mathbf{P}_{3,1} = (1, 0)$, $\mathbf{P}_{1,2} = (0.4, 1)$, $\mathbf{P}_{2,2} = (0.4, 0.4)$, $\mathbf{P}_{3,2} = (1, 0.4)$.	$\omega_{1,1} = 1$, $\omega_{2,1} = 1$, $\omega_{3,1} = 1$, $\omega_{1,2} = 1$, $\omega_{2,2} = 1$, $\omega_{3,2} = 1$.	$\xi = [0, 0, 0.5, 1, 1]$, $\eta = [0, 0, 1, 1]$.	[1,1]	80 × 40	[2, 2]	83 × 41	80 × 39
3D-CB	$\mathbf{P}_{1,1,1} = (0, 0, 0)$, $\mathbf{P}_{2,1,1} = (1.2, 0, 0)$, $\mathbf{P}_{1,2,1} = (0, 0.3, 0)$, $\mathbf{P}_{2,2,1} = (1.2, 0.3, 0)$, $\mathbf{P}_{1,1,2} = (0, 0, 0.6)$, $\mathbf{P}_{2,1,2} = (1.2, 0, 0.6)$, $\mathbf{P}_{1,2,2} = (0, 0.3, 0.6)$, $\mathbf{P}_{2,2,2} = (1.2, 0.3, 0.6)$.	$\omega_{1,1,1} = 1$, $\omega_{2,1,1} = 1$, $\omega_{1,2,1} = 1$, $\omega_{2,2,1} = 1$, $\omega_{1,1,2} = 1$ $\omega_{2,1,2} = 1$, $\omega_{1,2,2} = 1$, $\omega_{2,2,2} = 1$.	$\xi = [0, 0, 1, 1]$, $\eta = [0, 0, 1, 1]$, $\zeta = [0, 0, 1, 1]$.	[1,1,1]	40 × 5 × 20	[2, 2, 2]	41 × 6 × 21	39 × 4 × 19

Appendix C. Parameterization details

The quality of parameterization is crucial in both IGA and ITO. For simple and regular design domains, obtaining satisfactory parameterizations is relatively straightforward. In Table C.4, the parameterization details for several of the most common design domains are provided. In the numerical examples, the models are refined through a combination of degree elevation and knot insertion, with the new knots uniformly distributed within the unit interval. For the 3D bone model, its parameterization is generated using the approach outlined in [51], and we would like to express our gratitude to the authors for providing the parametrized model. For this model, $20 \times 10 \times 10$ knots are uniformly sampled in three parametric directions, the IGA mesh contains $19 \times 9 \times 9$ elements, and the total number of control points in the definition of NURBS volume is $21 \times 11 \times 11$, and the degrees of the basis functions are 2. Due to the large number of control points in the model, they are not listed in Table C.4. These data can be provided upon request.

Data availability

Data will be made available on request.

References

[1] M.P. Bendsoe, O. Sigmund, Topology Optimization: Theory, Methods, and Applications, Springer Science & Business Media, 2013.
[2] T.J. Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Engrg. 194 (39–41) (2005) 4135–4195, http://dx.doi.org/10.1016/j.cma.2004.10.008.
[3] J.A. Cottrell, T.J. Hughes, Y. Bazilevs, Isogeometric Analysis: Toward Integration of CAD and FEA, John Wiley & Sons, 2009, http://dx.doi.org/10.5555/1816404.
[4] H. Liu, X. Zhu, D. Yang, Isogeometric method based in-plane and out-of-plane free vibration analysis for Timoshenko curved beams, Struct. Eng. Mech. 59 (3) (2016) 503–526, http://dx.doi.org/10.12989/sem.2016.59.3.503.
[5] B. Hassani, M. Khanzadi, S.M. Tavakkoli, An isogeometrical approach to structural topology optimization by optimality criteria, Struct. Multidiscip. Optim. 45 (2012) 223–233, http://dx.doi.org/10.1007/s00158-011-0680-5.
[6] J. Gao, L. Gao, Z. Luo, P. Li, Isogeometric topology optimization for continuum structures using density distribution function, Internat. J. Numer. Methods Engrg. 119 (10) (2019) 991–1017, http://dx.doi.org/10.1002/nme.6081.
[7] J. Gao, L. Wang, Z. Luo, L. Gao, IgaTop: an implementation of topology optimization for structures using IGA in MATLAB, Struct. Multidiscip. Optim. 64 (2021) 1669–1700, http://dx.doi.org/10.1007/s00158-021-02858-7.

[8] Y. Wang, D.J. Benson, Isogeometric analysis for parameterized LSM-based structural topology optimization, Comput. Mech. 57 (2016) 19–35, http://dx.doi.org/10.1007/s00466-015-1219-1.
[9] J. Gao, M. Xiao, M. Zhou, L. Gao, Isogeometric topology and shape optimization for composite structures using level-sets and adaptive Gauss quadrature, Compos. Struct. 285 (2022) 115263, http://dx.doi.org/10.1016/j.compstruct.2022.115263.
[10] X. Xie, S. Wang, M. Xu, Y. Wang, A new isogeometric topology optimization using moving morphable components based on R-functions and collocation schemes, Comput. Methods Appl. Mech. Engrg. 339 (2018) 61–90, http://dx.doi.org/10.1016/j.cma.2018.04.048.
[11] W. Zhang, Q. Jiang, W. Feng, S.-K. Youn, X. Guo, Explicit structural topology optimization using boundary element method-based moving morphable void approach, Internat. J. Numer. Methods Engrg. 122 (21) (2021) 6155–6179, http://dx.doi.org/10.1002/nme.6786.
[12] Y. Wang, H. Xu, D. Pasini, Multiscale isogeometric topology optimization for lattice materials, Comput. Methods Appl. Mech. Engrg. 316 (2017) 568–585, http://dx.doi.org/10.1016/j.cma.2016.08.015.
[13] H. Liu, D. Yang, P. Hao, X. Zhu, Isogeometric analysis based topology optimization design with global stress constraint, Comput. Methods Appl. Mech. Engrg. 342 (2018) 625–652, http://dx.doi.org/10.1016/j.cma.2018.08.013.
[14] A. Gupta, B. Mamindlapelly, P.L. Karuthedath, R. Chowdhury, A. Chakrabarti, Adaptive isogeometric topology optimization using PHT splines, Comput. Methods Appl. Mech. Engrg. 395 (2022) 114993, http://dx.doi.org/10.1016/j.cma.2022.114993.
[15] P.L. Karuthedath, A. Gupta, B. Mamindlapelly, R. Chowdhury, A continuous field adaptive mesh refinement algorithm for isogeometric topology optimization using PHT-Splines, Comput. Methods Appl. Mech. Engrg. 412 (2023) 116075, http://dx.doi.org/10.1016/j.cma.2023.116075.
[16] P.L. Karuthedath, L. Barik, A. Gupta, A.K. Swain, R. Chowdhury, B. Mamindlapelly, A C1 continuous multi-patch framework for adaptive isogeometric topology optimization of plate structures, Comput. Methods Appl. Mech. Engrg. 429 (2024) 117132, http://dx.doi.org/10.1016/j.cma.2024.117132.
[17] X. Xie, S. Wang, M. Xu, N. Jiang, Y. Wang, A hierarchical spline based isogeometric topology optimization using moving morphable components, Comput. Methods Appl. Mech. Engrg. 360 (2020) 112696, http://dx.doi.org/10.1016/j.cma.2019.112696.
[18] J. Gao, X. Wu, M. Xiao, V.P. Nguyen, L. Gao, T. Rabczuk, Multi-patch isogeometric topology optimization for cellular structures with flexible designs using Nitsche's method, Comput. Methods Appl. Mech. Engrg. 410 (2023) 116036, http://dx.doi.org/10.1016/j.cma.2023.116036.
[19] W. Zhang, S. Jiang, C. Liu, D. Li, P. Kang, S.-K. Youn, X. Guo, Stress-related topology optimization of shell structures using IGA/TSA-based moving morphable void (MMV) approach, Comput. Methods Appl. Mech. Engrg. 366 (2020) 113036, http://dx.doi.org/10.1016/j.cma.2020.113036.
[20] W. Wang, D. Feng, L. Yang, S. Li, C.C. Wang, Topology optimization of self-supporting lattice structure, Addit. Manuf. 67 (2023) 103507, http://dx.doi.org/10.1016/j.addma.2023.103507.
[21] A. Garaigordobil, R. Ansola, J. Santamaría, I. Fernández de Bustos, A new overhang constraint for topology optimization of self-supporting structures in additive manufacturing, Struct. Multidiscip. Optim. 58 (2018) 2003–2017, http://dx.doi.org/10.1007/s00158-018-2010-7.
[22] J.P. Groen, J. Wu, O. Sigmund, Homogenization-based stiffness optimization and projection of 2D coated structures with orthotropic infill, Comput. Methods Appl. Mech. Engrg. 349 (2019) 722–742, http://dx.doi.org/10.1016/j.cma.2019.02.031.
[23] Y. Liu, M. Zhou, C. Wei, Z. Lin, Topology optimization of self-supporting infill structures, Struct. Multidiscip. Optim. 63 (2021) 2289–2304, http://dx.doi.org/10.1007/s00158-020-02805-y.
[24] C. Wang, X. Qian, Simultaneous optimization of build orientation and topology for additive manufacturing, Addit. Manuf. 34 (2020) 101246, http://dx.doi.org/10.1016/j.addma.2020.101246.
[25] M. Bruggi, V. Laghi, T. Trombetti, Simultaneous design of the topology and the build orientation of wire-and-arc additively manufactured structural elements, Comput. Struct. 242 (2021) 106370, http://dx.doi.org/10.1016/j.compstruc.2020.106370.
[26] J. Liu, H. Yu, Concurrent deposition path planning and structural topology optimization for additive manufacturing, Rapid Prototyp. J. 23 (5) (2017) 930–942, http://dx.doi.org/10.1108/RPJ-05-2016-0087.
[27] M. Boissier, G. Allaire, C. Tournier, Concurrent shape optimization of the part and scanning path for powder bed fusion additive manufacturing, SIAM J. Control Optim. 61 (2) (2023) 697–722, http://dx.doi.org/10.1137/21M1461976.
[28] W. Wang, Y. Xia, Topology optimization based channel design for powder-bed additive manufacturing, Addit. Manuf. 54 (2022) 102717, http://dx.doi.org/10.1016/j.addma.2022.102717.
[29] Q. Li, W. Chen, S. Liu, L. Tong, Structural topology optimization considering connectivity constraint, Struct. Multidiscip. Optim. 54 (2016) 971–984, http://dx.doi.org/10.1007/s00158-016-1459-5.
[30] C. Dapogny, R. Estevez, A. Faure, G. Michailidis, Shape and topology optimization considering anisotropic features induced by additive manufacturing processes, Comput. Methods Appl. Mech. Engrg. 344 (2019) 626–665, http://dx.doi.org/10.1016/j.cma.2018.09.036.
[31] K. Wu, W. Wang, F. van Keulen, J. Wu, Space–time topology optimization for anisotropic materials in wire and arc additive manufacturing, Int. J. Mech. Sci. (2024) 109712, http://dx.doi.org/10.1016/j.ijmecsci.2024.109712.
[32] C. Dai, C.C. Wang, C. Wu, S. Lefebvre, G. Fang, Y.-J. Liu, Support-free volume printing by multi-axis motion, ACM Trans. Graph. 37 (4) (2018) 1–14, http://dx.doi.org/10.1145/3197517.3201342.
[33] C. Wu, C. Dai, G. Fang, Y.-J. Liu, C.C. Wang, RoboFDM: A robotic system for support-free fabrication using FDM, in: 2017 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2017, pp. 1175–1180, http://dx.doi.org/10.1109/ICRA.2017.7989140.
[34] T.A. Rodrigues, V. Duarte, R. Miranda, T.G. Santos, J. Oliveira, Current status and perspectives on wire and arc additive manufacturing (WAAM), Mater. 12 (7) (2019) 1121, http://dx.doi.org/10.3390/ma12071121.
[35] G. Fang, T. Zhang, S. Zhong, X. Chen, Z. Zhong, C.C.L. Wang, Reinforced FDM: multi-axis filament alignment with controlled anisotropic strength, 39 (6) (2020). http://dx.doi.org/10.1145/3414685.3417834.
[36] W. Wang, D. Munro, C.C. Wang, F. van Keulen, J. Wu, Space-time topology optimization for additive manufacturing: Concurrent optimization of structural layout and fabrication sequence, Struct. Multidiscip. Optim. 61 (2020) 1–18, http://dx.doi.org/10.1007/s00158-019-02420-6.
[37] W. Wang, F. van Keulen, J. Wu, Fabrication sequence optimization for minimizing distortion in multi-axis additive manufacturing, Comput. Methods Appl. Mech. Engrg. 406 (2023) 115899, http://dx.doi.org/10.1016/j.cma.2023.115899.
[38] S. Liu, Q. Li, W. Chen, L. Tong, G. Cheng, An identification method for enclosed voids restriction in manufacturability design for additive manufacturing structures, Front. Mech. Eng. 10 (2015) 126–137, http://dx.doi.org/10.1007/s11465-015-0340-3.
[39] Y. Luo, O. Sigmund, Q. Li, S. Liu, Additive manufacturing oriented topology optimization of structures with self-supported enclosed voids, Comput. Methods Appl. Mech. Engrg. 372 (2020) 113385, http://dx.doi.org/10.1016/j.cma.2020.113385.
[40] W. Wang, K. Wu, F. van Keulen, J. Wu, Regularization in space-time topology optimization for additive manufacturing, Comput. Methods Appl. Mech. Engrg. 431 (2024) 117202, http://dx.doi.org/10.1007/s00158-019-02420-6.
[41] L. Piegl, W. Tiller, The NURBS Book, Springer Science & Business Media, 2012, http://dx.doi.org/10.1007/978-3-642-97385-7.
[42] D. Shepard, A two-dimensional interpolation function for irregularly-spaced data, in: Proceedings of the 1968 23rd ACM National Conference, 1968, pp. 517–524, http://dx.doi.org/10.1145/800186.810616.
[43] F. Wang, B.S. Lazarov, O. Sigmund, On projection methods, convergence and robust formulations in topology optimization, Struct. Multidiscip. Optim. 43 (2011) 767–784, http://dx.doi.org/10.1007/s00158-010-0602-y.
[44] T.J. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Courier Corporation, 2003.
[45] K. Svanberg, The method of moving asymptotes—a new method for structural optimization, Internat. J. Numer. Methods Engrg. 24 (2) (1987) 359–373, http://dx.doi.org/10.1002/nme.1620240207.

[46] T. Sakai, On Riemannian manifolds admitting a function whose gradient is of constant norm, Kodai Math. J. 19 (1) (1996) 39–51, http://dx.doi.org/10.2996/kmj/1138043545.
[47] T. Zhang, G. Fang, Y. Huang, N. Dutta, S. Lefebvre, Z.M. Kilic, C.C. Wang, S3-slicer: A general slicing framework for multi-axis 3D printing, ACM Trans. Graph. 41 (6) (2022) 1–15, http://dx.doi.org/10.1145/3550454.3555516.
[48] T. Liu, T. Zhang, Y. Chen, Y. Huang, C.C. Wang, Neural slicer for multi-axis 3D printing, ACM Trans. Graph. 43 (4) (2024) 1–15, http://dx.doi.org/10.1145/3658212.
[49] C. De Boor, On calculating with B-splines, J. Approx. Theory 6 (1) (1972) 50–62, http://dx.doi.org/10.1016/0021-9045(72)90080-9.
[50] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, Adv. Comput. Math. 4 (1995) 389–396, http://dx.doi.org/10.1007/BF02123482.
[51] H. Lin, S. Jin, Q. Hu, Z. Liu, Constructing B-spline solids from tetrahedral meshes for isogeometric analysis, Comput. Aided Geom. Design 35 (2015) 109–120, http://dx.doi.org/10.1016/j.cagd.2015.03.013.