

# Solid isotropic material with thickness penalization – A 2.5D method for structural topology optimization



Tejeswar Yarlagadda <sup>a,\*</sup>, Zixin Zhang <sup>a,b</sup>, Liming Jiang <sup>a,b</sup>, Pradeep Bhargava <sup>c</sup>, Asif Usmani <sup>a,b</sup>

<sup>a</sup> Department of Building Environment and Energy Engineering, The Hong Kong Polytechnic University, Hong Kong Special Administrative Region

<sup>b</sup> Research Institute for Sustainable Urban Development, The Hong Kong Polytechnic University, Hong Kong Special Administrative Region

<sup>c</sup> Structural Engineering Group, Civil Engineering Department, Indian Institute of Technology, Roorkee, India

## ARTICLE INFO

### Article history:

Received 12 July 2021

Accepted 24 June 2022

Available online 4 July 2022

### Keywords:

Topology optimization

Nodal design variable

Degenerated element

2.5D element

SIMP

SIMTP

## ABSTRACT

SIMP is the most common topology optimization scheme to minimize the material utilization of a structural component using element densities as a design variable. This paper presents a new methodology, SIMTP, Solid Isotropic Material with Thickness Penalization, introducing a nodal thickness variable to the plane stress element. SIMTP uses a 2.5D element, and it is developed considering the varying thickness as a 3D problem and planar transformation as a 2D problem. The developed 2.5D element is used for projecting the 2D strain energy onto a 3D space. Classical optimization problems like Cantilever, MBB, and L-, beams are solved using 2.5D SIMTP, and results are compared with SIMP. The implementation is simple, directly representing the thickness, and nodal design variables enhance the resolution. Checkerboarding or other topology-related issues are not noticed during the process. Also, 2.5D SIMTP can yield desired results with fewer elements, thereby reducing the computational efforts. Besides, a 211-line MATLAB implementation of 2.5D SIMTP with examples is provided at the end of this paper.

© 2022 Elsevier Ltd. All rights reserved.

## 1. Introduction

Construction oriented optimization is gaining traction with newly emerging technologies [1–2], especially in the context of additive manufacturing processes. Various mathematical formulations have been developed to optimize the size, shape, and topology of a component. Early in the 1980s, optimization was limited to sizing (cross-section) and shaping (geometry), making it challenging to generate the holes in topology for various reasons. Topology optimization (TO) methods are the first to introduce holes in a design space by gradually removing the material. Several TO methods are available in the literature [3–14], and Solid Isotropic Material/Microstructure with Penalization [4–6]. SIMP is the widely used and well-accepted standard tool for optimizing lightweight components. SIMP has been used in the past two decades to optimize the structural components through the finite element method (FEM) and is implemented in most commercial software due to its simplicity and efficiency. A conventional finite element method (FEM) yields stiffness-based solutions under a given load and boundary conditions. Density as a design variable conventional FEM delivers stable solutions until the material remains within its elastic limits. However, resulting low-density design values from

the optimization process with material non-linearity cause the loss of positive definiteness of the stiffness matrix yielding unstable solutions. Material non-linearity in TO is dealt with in several ways available in the literature, and the most popular are element removal methods like ESO [15]. Apart from non-linearity, TO problems may suffer from checkerboarding, mesh dependency, point flexure/hinge formation (one-node and de-facto) in compliant mechanisms, pixelated (low resolution) images, grayscaling, and thin member formations. There are many alternatives in the literature to avoid the issues mentioned above, such as (a) using higher-order elements [16–17]; (b) regularization techniques implemented either by filtering or adding extra constraints such as sensitivity filter [18–19], density filter [20–21], MOLE constraint method [22], perimeter control [23], slope control [24] and minimum member size control [25]; (c) projection methods such as Heaviside filter [26] and Morphology filter [27]; (d) non-conforming finite elements [28–29] and adaptive refinement/mesh adjustment [30–33].

TO pathologies, as mentioned earlier, are primarily associated with using element-based density variables (EDV). In a continuous effort to enhance TO, nodal density variables (NDV) were introduced by Kumar and Gossard [34]. As cited in the monography of Bendsøe and Sigmund [35], Hammer [36] suggested using NDV to suppress checkerboarding. However, Bendsøe and Sigmund [35] reported that the method was undesirable and resulted in

\* Corresponding author.

E-mail address: [tejeswar.yarlagadda@connect.polyu.hk](mailto:tejeswar.yarlagadda@connect.polyu.hk) (T. Yarlagadda).

zig-zag boundaries. However, Belytschko et al. [37], Matsui and Terada [38], and Guest et al. [26,39–40] reported no checkerboarding or any other TO pathologies after using NDV. Nevertheless, Rahmatalla and Swan [41] found the so-called layering or islanding phenomenon while using NDV on Q4 elements with coarse meshes. The term “islanding” is not new in the engineering field, especially in electric grid power systems, and in structural topology optimization (STO), it refers to the unwanted placing of material in a layered manner within the design domain. Paulino and Le [42] used an internal averaging technique that successfully eliminated the islanding. Kang and Wang [43,44] successfully demonstrated the elimination of islanding, mesh-dependency, and checkerboarding using NDV with higher-order elements (Q8). Also, Kang and Wang used local [44] and non-local [43] Shepard interpolants to interpolate the nodal densities since interpolation using regular Q8 shape functions may cause negative density values. Moreover, non-local Shepard interpolants introduced in [43] have the flexibility of separating the design variable points from the finite element mesh [32] and thus can be positioned at freely chosen points other than element nodes, facilitating design variable refinement to achieve a higher quality of the boundary description. Like NDV methods, Nguyen et al. [45] proposed a multiresolution topology optimization method (MTOP) to produce high-resolution images and reduce computational costs. More studies are listed here on the TO using NDV in conjunction with, functionally graded materials [46], classical Newton-Raphson solution [47], adaptive mesh refinement [30–33,40,48–49], Isogeometric analysis [50–52], meshfree methods [53–59], virtual element method [60], stress constrained TO [61], geometric non-linearity [53,62], ESO [63], compliant mechanisms [58,64–67], design-dependent surface loading [68–69], piezoelectric actuators [70–71], MTOP [72]. Following review articles are suggested, STO [73–76], Level set methods [77], Aircraft and Aerospace structures [78], Compliant mechanisms [79] and Additive manufacturing [80], to cover the best knowledge on TO, its approaches, applications, issues, and manufacturing possibilities and constraints.

This paper presents a new optimization methodology called SIMTP, including a 2.5D plane stress element. SIMTP uses thickness as a nodal design variable, similar to the variable thickness sheet problem. The thickness design variable is not a new idea in optimization. Rossow and Taylor [81] proposed variable thickness sheet optimization but failed to generate holes since the thickness is a dimensional variable and cannot be penalized, which is the primary advantage of the TO methods. The thickness optimization background until the 1990s and its examples were published in Bendsøe [82]. Bendsøe [4] also stated that the basic formulation of the SIMP excluding penalty represents the variable thickness sheet problem. Li et al. [83,84] proposed a thickness-based ESO where the material with lower thickness was systematically removed to create the holes. Li et al. [85] also proposed a smoothing algorithm to avoid checkerboarding in the ESO and demonstrated a few examples. Like the variable thickness sheet, Makrodimopoulos et al. [86,87] presented thickness optimization on plane stress element formulation by evaluating thickness at Gauss points, while the optimal thicknesses obtained failed to produce the holes. Kennedy [88] proposed the discrete thickness optimization (DTO) method that converts discrete thickness into a continuous variable using a non-dimensional interpolation parameter similar to the density variable in the TO. This interpolation parameter with intermediate designs was then penalized piecewise and subjected to a constraint function based on SIMP/RAMP. The cases presented in the above study are promising, and the formulation could generate holes.

The thickness optimization proposed in the present paper is similar to SIMP [4] and Kennedy's DTO [88]. SIMTP penalizes a dimensionless parameter called the thickness factor to scale the thickness at a point. The primary advantage of the SIMTP over other sizing problems lies in its fundamental formulation of freeing the constitutive matrix from the design variable. Design variable independent of the constitutive matrix is achieved by developing a 2.5D element formed by degenerating a 3D twenty-noded (serendipity) element to a 2D eight-noded element holding the plane stress assumptions. The above process seeks the projection of 2D strain energy onto a 3D volume, justifying the term 2.5D. Thus a volume based on the stable 3D Jacobian ensures the numerical stability in 2.5D SIMTP. Further, this process does not interfere with the traditional non-linear FEM.

The rest of the paper is organized into five sections and is as follows, (a) [Section 2](#) presents the formulation of a 2.5D plane stress element with a patch test, (b) [Section 3](#) introduces SIMTP as a thickness optimization problem, (c) [Section 4](#) demonstrates MBB, Cantilever, and L-, beams optimum thickness distribution based on the proposed method and its comparison with the SIMP, (d) [Section 5](#) gives the final remarks on the present study.

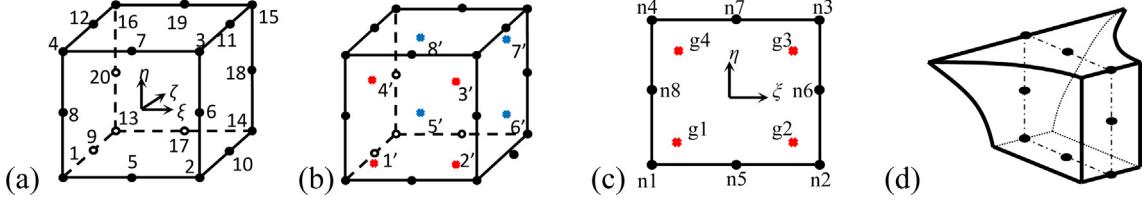
## 2. Implementation of 2.5D element

### 2.1. 2.5D element

Flexural structural components such as beams are commonly designed for in-plane loading using plane stress assumptions. Under such loading conditions, there is little or no influence of out of plane stress on the integrity of the component, as that is usually taken care of through measures such as diaphragms connecting steel or RC beams together to form a grillage for a bridge superstructure. For such situations, which are potentially the majority of design cases in practice, 2D plane stress assumptions are perfectly adequate. There is a question of stability in the third dimension if the optimized structure results in slender portions under compressive stress. This issue can be resolved by enforcing appropriate constraints. The present study uses nodal thicknesses as design variables for the optimization, but a 2D plane stress element is generally based on a unit thickness. Therefore, concerning the 2D state of stress and 3D volume, the 2.5D element is formulated by degenerating the 3D serendipity element, as illustrated in [Fig. 1](#). The quadratic serendipity element is used to take advantage of higher-order elements that can partially avoid numerical instabilities in TO [17,43–44].

In [Fig. 1](#), third-dimension  $\zeta$  represents the thickness direction ( $z$ -direction),  $\xi$  and  $\eta$  represent the planar directions  $x$  and  $y$ , respectively,  $1 - 20$  and  $n1 - n8$  represent the node numbering, and  $1' - 8'$  and  $g1' - g4'$ , represent the sampling point numbering (Gaussian quadrature). Assuming,

- I. No element distortion in planar directions, i.e., planar coordinates of the nodes along the thickness direction, remain the same (refer to [Fig. 6](#));
- II. Nodal deformations across the thickness are equal;
- III. Plane stress assumptions remain valid, but the volume of an element is based on the nodal thickness;
- IV. Quadrature rule is maintained throughout the process (e.g., two-point integration rule is used in this paper, i.e.  $2 \times 2$  for planar transformation,  $2 \times 2 \times 2$  for volume estimation);



**Fig. 1.** Degeneration of 3D serendipity and 2.5D element formulation. (a)3D element; (b) Quadrature rule; (c) 2.5D planar transformation; (d) 2.5D variable nodal thickness.

Planar assumptions:

$$\begin{aligned} \forall(i,j) \in & \left\{ \begin{array}{l} (\{1, 9, 13\}, n1) \\ (\{2, 10, 14\}, n2) \\ (\{3, 11, 15\}, n3) \\ (\{4, 12, 16\}, n4) \\ (\{5, 17\}, n5) \\ (\{6, 18\}, n6) \\ (\{7, 19\}, n7) \\ (\{8, 20\}, n8) \end{array} : (x_i, y_i) = (x_j, y_j); (u_i, v_i) = (u_j, v_j) \right. \\ \forall(i,j) \in & \left\{ \begin{array}{l} (1, n1) \\ (2, n2) \\ (3, n3) \\ (4, n4) \\ (5, n5) \\ (6, n6) \\ (7, n7) \\ (8, n8) \end{array} : \begin{cases} z_i = t_j/2 \\ z_{i+12} = -t_j/2 \end{cases} \right. \\ \forall i \in & \{9, 10, 11, 12\} : z_i = 0 \end{aligned} \quad (1)$$

Degenerating the 3D element using the assumptions in Eq. (1) leads to the 2.5D element shape functions for the planar transformation as follows,

$$\begin{aligned} N_{ni}^{2.5D} &= N_i^{3D} + \beta N_{i+8}^{3D} + N_{i+12}^{3D} \\ \forall i \in Z_{[1,4], [13,16]} &: N_i^{3D} \\ &= \frac{1}{8}(1 + \xi_i \xi)(1 + \eta_i \eta)(1 + \zeta_i \zeta)(\xi_i \xi + \eta_i \eta + \zeta_i \zeta - 2) \\ \forall i \in \{5, 7, 17, 19\} &: N_i^{3D} = \frac{1}{4}(1 - \xi^2)(1 + \eta_i \eta)(1 + \zeta_i \zeta) \\ \forall i \in \{6, 8, 18, 20\} &: N_i^{3D} = \frac{1}{4}(1 + \xi_i \xi)(1 - \eta^2)(1 + \zeta_i \zeta) \\ \forall i \in \{9, 10, 11, 12\} &: N_i^{3D} = \frac{1}{4}(1 + \xi_i \xi)(1 + \eta_i \eta)(1 - \zeta^2) \\ \text{suchthat} : & \begin{cases} \forall i \in Z_{[1,4]} : \beta = 1 \\ \forall i \in Z_{[5,8]} : \beta = 0 \\ \xi, \eta, \zeta \in [-1, 1] \end{cases} \end{aligned} \quad (5)$$

The elemental stiffness matrix is then given by.

$$K_{e,j} = \left( \sum_{l=1}^{n_g} \sum_{m=1}^{n_g} \sum_{n=1}^{n_g} \underbrace{[B(\xi_l, \eta_m)]^T [D] [B(\xi_l, \eta_m)]}_{2D} \underbrace{|J(\xi_l, \eta_m, \zeta_n)| w_l w_m w_n}_{3D} \right) \quad (6)$$

Thin-member assumptions for in-plane loading:

$$\sigma_{zz} = \sigma_{xz} = \sigma_{yz} = 0 \quad (2)$$

Where  $(x_i, y_i, z_i)$  is the coordinate of the  $i^{th}$  node in the 3D element, and  $(x_j, y_j)$  and  $t_j$  are coordinate and thickness of the  $j^{th}$  node in the 2.5D element, respectively.  $(u_i, v_i)$  and  $(u_j, v_j)$  represent the nodal deformations of  $i^{th}$  node in 3D and  $j^{th}$  node in 2.5D, respectively, while  $u$  and  $v$  are deformations in  $x$  and  $y$  directions,  $(i,j)$  sets in the Eq. (1) refer to the node numbering in Fig. 1. The above assumptions in Eq. (1) lead the resulting shape functions as follows.

Geometry in a 3D serendipity element is.

$$\alpha = \sum_{i=1}^{20} N_i^{3D} \alpha_i \quad (3)$$

Planar transformation in 2.5D element is given by.

$$\alpha = \sum_{i=1}^8 N_{ni}^{2.5D} \alpha_{ni} \quad (4)$$

Where,  $\alpha$  is a general representation for coordinates  $(x, y, z)$  or deformations  $(u, v)$ ,  $N_i^{3D}$  and  $N_{ni}^{2.5D}$  are shape functions of  $i^{th}$  node in a 3D and 2.5D element, respectively,  $(\xi_i, \eta_i, \zeta_i)$  is an isoparametric coordinate of the  $i^{th}$  node in an element,  $w_l, w_m, w_n$  are Gauss weights of the sample points  $(l, m, n)$ . The formulated 2.5D shape functions are similar to the 2D serendipity shape functions. 2.5D (Eq. (4)) and 3D (Eq. (3)) shape functions generate strain energy and volumetric terms, respectively, to form the stiffness matrix for the 2.5D element, refer to Eq. (6). Similarities can be noticed between the proposed 2.5D element and the shell element with varying nodal thicknesses [89–93]. Shell elements, in general, contain rotational degrees of freedom (DoF) additional to translations and are often characterized as thick (6 DoF's) considering drilling rotation else thin (5 DoF's), where the proposed 2.5D element is limited to in-plane translational degrees of freedom. Thin shell elements with in-plane loading may have a similar outcome as the present 2.5D element since out-of-plane stresses are absent. However, shell elements are susceptible to locking and require more computational energy.

## 2.2. Element stability and accuracy

Serendipity elements are less susceptible to locking. Further, using more elements and reduced quadrature improves the results [94]. Literature also suggests that the serendipity element should be used with care because its robustness is questionable in the following cases,

- i. Reduced quadrature – possible spurious mode occurrence;
- ii. Non-linear problems (large-displacement, contact analyses) – possible spurious mode occurrence;
- iii. Single element – possible spurious mode occurrence;
- iv. Distorted element – leads to inaccurate results;

However, the current problem is limited to the linear elastic case with a sufficient number of elements, but the out of the plane distortion is unavoidable considering the nodal thickness variable. Therefore, it is suggested to test the behavior of the above formulated 2.5D element. Two cases illustrated in Fig. 2 are studied under a linear elastic case with (a) case-I: linearly and (b) case-II: non-linearly varying thicknesses. 3D models of these two cases are built in ABAQUS 6.14 to compare the results with the 2.5D element implemented in MATLAB.

The two test cases have the following properties, (a) top surface load of 0.1 N/m<sup>2</sup> for case-I and 0.16 N/m<sup>2</sup> for case-II; (b) maximum thickness of 0.1 m; (c) elastic modulus of 2500 N/m<sup>2</sup>; (d) zero Poisson ratio and (e) simply supported boundary conditions. Surface load is applied on both 2.5D and 3D models since line loads cannot be used on a 3D model. Uniformly varying load (UVL) equivalent to surface load is applied on the 2.5D models as solved in Eq. (7–9) and refer to Fig. 3. The whole material space lies in the prismatic domain of 1.0 m × 0.1 m × 0.1 m, and

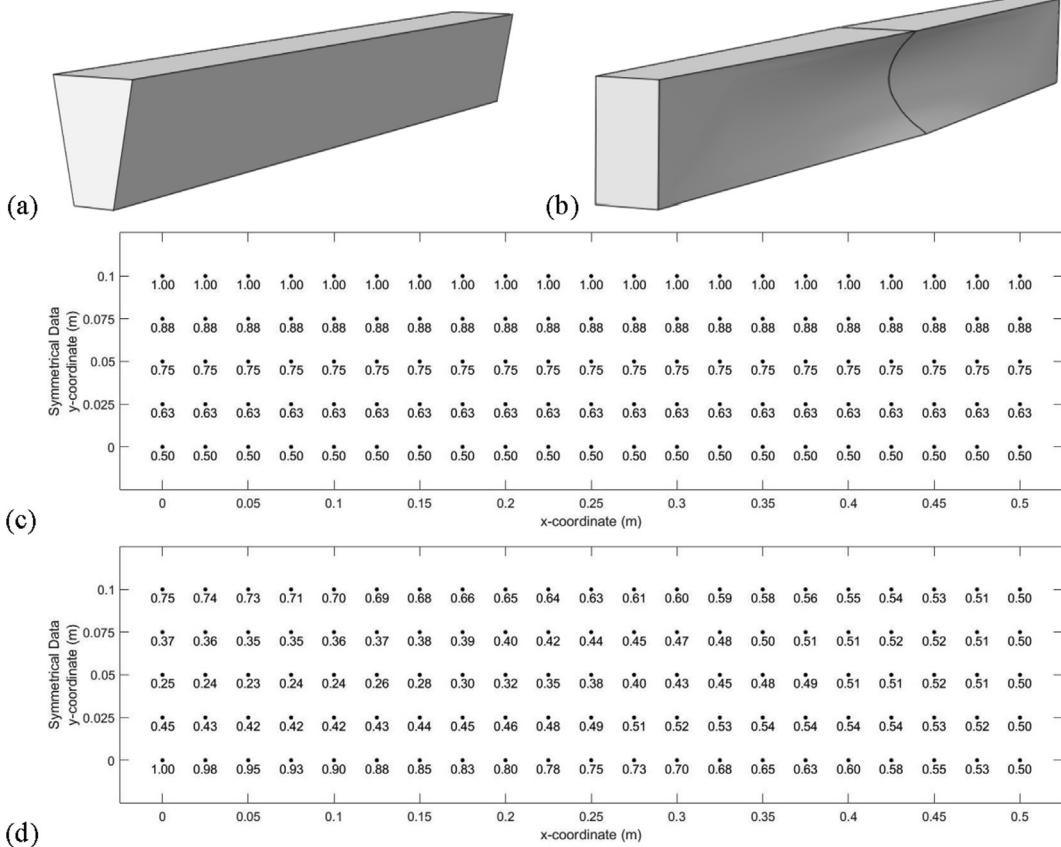


Fig. 2. 3D models and thickness data of the test cases. (a) Case-I: linear thickness model; (b) Case-II: non-linear thickness model; (c) Surface data for case-I; (d) Surface data for case-II.

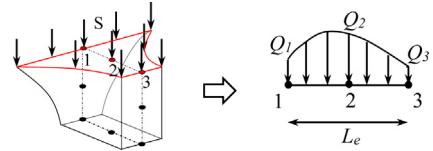


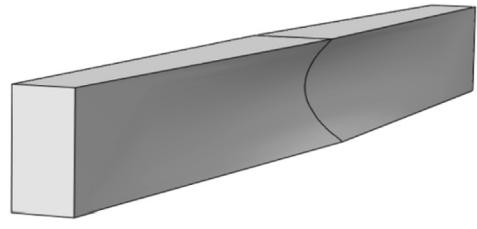
Fig. 3. Surface load as UVL.

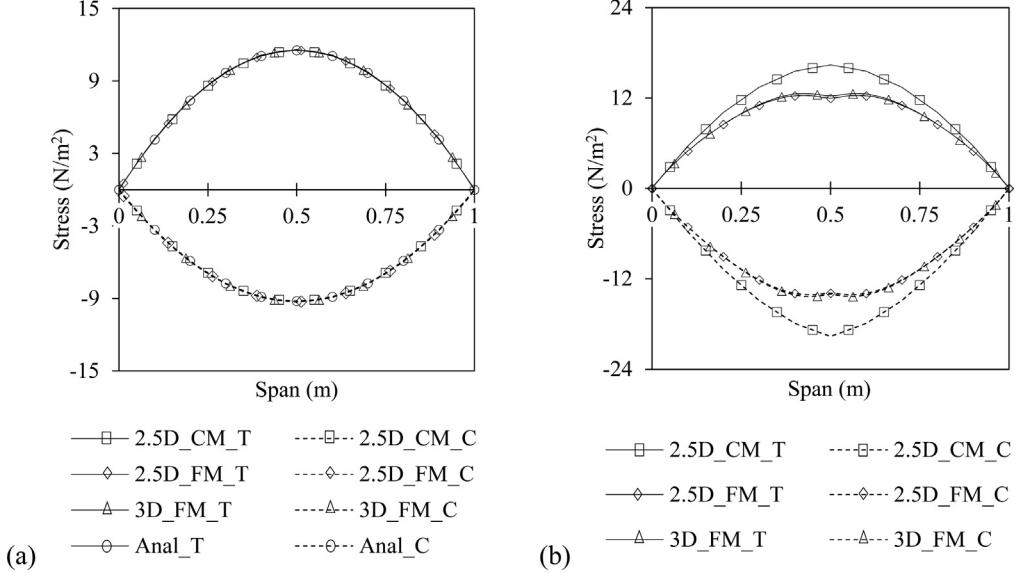
the symmetrical surface data (factored thickness) for the two test cases are presented below in Fig. 2(c) and 2(d), where planar coordinates (0,0) and (0.5,0.1) represent the bottom mid-span and the top right boundary of the beam, respectively. Nodal thicknesses are obtained by multiplying the surface data provided in Fig. 2 with the base thickness of 0.1 m. The surface data, presented in Fig. 2, is for the corner nodes, while mid-node thicknesses can be estimated as the average of two adjacent nodes. Full-scale models were analyzed, but symmetrical data was provided to avoid a clumsy representation.

Eq. (7–9) presents the conversion of the surface load  $S$  (highlighted as red in Fig. 3) acting on a 2.5D element to concentrated forces on edge nodes assuming edge as a linear element. Firstly,  $S$  is converted to UVL, and UVL at any point on the edge is given by.

$$Q = L_1 Q_1 + L_2 Q_2 + L_3 Q_3 = [L_1 \quad L_2 \quad L_3] \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} = L \left( S \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \right) = LP \quad (7)$$

Where  $Q_1, Q_2$  and  $Q_3$  are obtained by the product of  $S$  and corresponding nodal thicknesses  $t_1, t_2$  and  $t_3$ , respectively. Load  $Q$  (UVL) is then distributed as the concentrated load on nodal points as follows.



**Fig. 4.** Comparison of stresses at top and bottom fibers. (a) Case-I and (b) Case-II.**Table 1**  
Comparison of midspan downward deflections.

| Case    | 2.5D (CM) | 2.5D (FM) | 3D (FM) | Analytical |
|---------|-----------|-----------|---------|------------|
| I (mm)  | 8.791     | 8.813     | 8.816   | 8.804      |
| II (mm) | 14.469    | 11.544    | 11.247  | -          |

$$[F] = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = \int_{-1}^1 L^T LP \partial x / \partial \xi d\xi$$

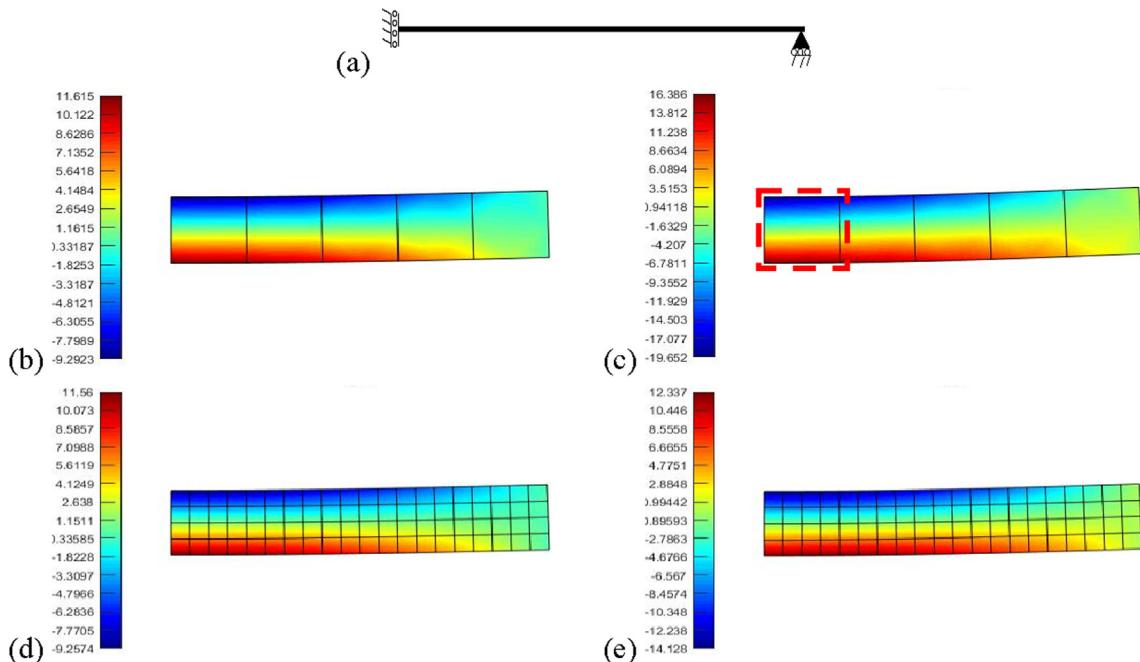
$$x = L_1 x_1 + L_2 x_2 + L_3 x_3 \quad (8)$$

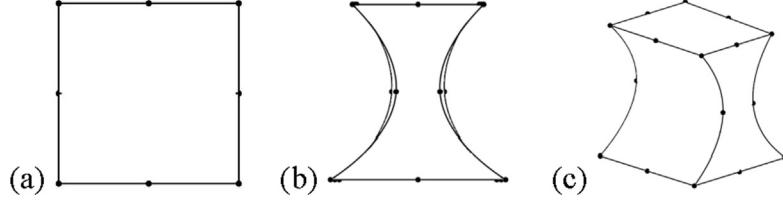
The force vector is obtained from solving the above integration.

$$F = \begin{bmatrix} 4 & 2 & -1 \\ 2 & 16 & 2 \\ -1 & 2 & 4 \end{bmatrix} PL_e / 30 \quad (9)$$

$F$  is the force vector of the element edge,  $S$  is the surface load,  $L_e$  is the edge length of the element,  $L$  is the shape function matrix of the quadratic three-noded line element,  $P$  is the UVL matrix.

In Fig. 4, CM and FM are coarse and fine mesh, respectively, T and C are tension in bottom fiber and compression in top fiber, respectively, Anal is analytical. Coarse-mesh of  $10 \times 1$ , and fine-mesh of  $40 \times 4$ , are used in 2.5D models to identify possible instabilities and the accuracy of the element. 3D model cannot be discretized using the above coarse mesh due to resulting instabilities. Therefore, it is modeled with a fine discretization of  $40 \times 4 \times 3$ . MATLAB and ABAQUS results for both test cases are

**Fig. 5.** Deformed stress contours in 2.5D models. (a) Results are shown for the right half beam; (b) Case-I: CM; (c) Case-II: CM; (d) Case-I: FM and (e) Case-II: FM.



**Fig. 6.** Element's profile as highlighted in Fig. 5(c). (a) Front (+XY); (b) Side (+ZY) and (c) Isometric, views.

shown in Fig. 4 and summarized in Table 1. An analytical solution based on 2D elasticity is provided for Case I in Appendix A. Due to the non-linear thickness across the span, the analytical solution for Case II is too difficult to be given. Besides, deformed stress contours are presented in Fig. 5 for the right half beam of both 2.5D cases. It is evident from the results that 2.5D elements can produce stable and accurate results for both cases. Although accuracy is achievable with fewer elements, coarse mesh might produce unreliable results for extremely varying thicknesses, evident from Fig. 4(b).

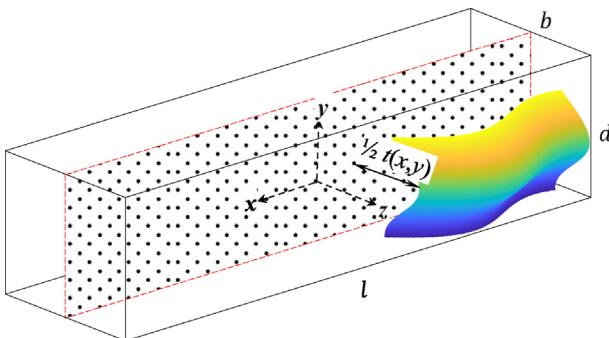
### 2.3. The computational efficiency of the 2.5D element

Midspan downward deflection results for case-I in Table 1 show that the 2.5D element results correspond to 3D fine mesh and analytical results. Case-II with coarse mesh showed flexible results relative to its alternative model meshes. In general, coarse meshes yield stiffer results, which is also evident in case-I, where the cross-section along the span is uniform in case-I while varying in case-II. So, it should be noted that the 2.5D elements are different from traditional 2D elements, where the behavior of 2.5D elements is greatly influenced by extreme thickness variation within the element. Further, to understand any such influence, the geometry for an element highlighted in Fig. 5(c) is presented in Fig. 6, and its surface data can be extracted from Fig. 2(d). It can be observed that the element is squeezed at mid-height relative to the top and bottom, indicating poor discretization, which often leads to computationally inefficient results and is the cause for yielding flexible results. It is also noticed that Jacobian is positive and stable in this case, despite the computational efficiency. So stable Jacobian and mesh quality are equally important in achieving accurate results. Therefore, it is suggested to use more 2.5D elements to avoid extreme thickness variation within the element and improve the computational efficiency.

## 3. SIMTP and thickness optimization formulation

### 3.1. Definition of SIMTP

SIMTP assumes a solid isotropic material occupying a domain  $\Omega^m$ , symmetrical to the midplane, as highlighted in Fig. 7. The above material domain lying in a prismatic design space  $\Omega$  of



**Fig. 7.** Boundary representation of a 3D solid body.

dimensions  $l \times b \times d$  is represented as lumped point masses. These point masses are allowed to move only in the symmetrical plane, as illustrated in Fig. 7. Any point mass (PM) at  $(x,y)$  is having a characteristic thickness of  $t(x,y)$ . The integral of these point mass thicknesses over the symmetrical plane represents the material boundary of the body. Initially, this boundary can be a parametric surface/an aesthetical and architectural expression. Thus, for any slightest movement of PM under relative external/internal influences (e.g., deformations of a node in finite element space),  $t(x,y)$  is a continuous variable and is defined as.

$$\forall(x,y) \in \Omega : t(x,y) = [f_{\min} + f(x,y)^p(f_{\max} - f_{\min})]b$$

$$\text{suchthat : } \begin{cases} f(x,y) \in [0, 1]^R \\ p \in R_{\geq 1} \\ \Omega^m \subseteq \Omega_R^3 \end{cases} \quad (10)$$

Where  $f(x,y)$  is a dimensionless parameter called thickness-factor (TF) and  $f_{\max}, f_{\min}$  are maximum and minimum TF's. Penalty ( $p$ ) is used to avoid intermediate thickness values. The penalty is taken as a unit value for the initial geometry, and TF is determined with reference to the initially approximated boundary from Eq. (10). Like the density variable in TO, TF is also restricted to positive unit interval values with minimum thickness to avoid the singular mass system (i.e., PM undergoes an infinite movement under any infinitely smaller influence). Also, a unit value of TF indicates PM with a total thickness of  $b$ , while zero is considered void. Here penalizing the TF does not change the dimensionality of the thickness, ensuring consistency and continuity. The above-discussed methodology is named Solid Isotropic Material with Thickness Penalization, in short SIMTP.

### 3.2. Thickness optimization problem

The optimization problem is solved using the finite element approximation such that every node represents PM in SIMTP, and nodal thicknesses are the obvious design candidates to evaluate the minimum compliance and sensitivity analysis.

#### 3.2.1. Minimum compliance

The minimum compliance is achieved by minimizing the strain energy for a given set of constraints, and it is obtained by regularizing the element's stiffness by varying the thickness, indeed TF.

$$\min : C = u^T K u$$

$$\text{subjectto : } \begin{cases} Ku = F \\ V/V_o \leq V_r \\ 0 \leq f \leq 1 \end{cases}$$

$$K = \sum_{j \in N_e} K_{e,j}$$

$$V = \sum_{j \in N_e} V_{e,j} \quad (11)$$

Where  $K$  is an assembly of elemental stiffness matrices corresponding to joint DoF's,  $K_{e,j}$  is the  $j^{th}$  element stiffness matrix esti-

mated based on nodal TF,  $N_e$  is the whole set of elements,  $u$  is nodal deformations matrix (movement of the point mass) under external loads or body forces  $F$  subjected to boundary conditions,  $V$  is the volume of the total body mass for an obtained or a given set of TF's,  $V_0$  is a volume of the entire body mass for initially approximated TF's,  $V_r$  is the volume ratio limit,  $V_{e,j}$  is  $j^{th}$  element volume estimated based on nodal TF's,  $f$  is TF as a design variable.

### 3.2.2. Sensitivity analysis

Sensitivity analysis is carried out at the nodal level since the design candidates are nodal variables. It should be noted that the node is a nonintegral variable. Therefore, the stiffness rate for the sensitivity analysis is evaluated at the element level and lumped to the nodes. And the thickness at any point inside an element space ( $\Omega_e$ ) is interpreted as follows,

$$\forall(x,y) \in \Omega_e : t(x,y) = \sum_{i \in N_n, n_e} w_i t_i$$

$$w_i = \frac{1 / ((x - x_i)^2 + (y - y_i)^2)}{\sum_{i \in N_n, n_e} 1 / ((x - x_i)^2 + (y - y_i)^2)} \quad (12)$$

Where  $n_e$  is the set of nodes in an element,  $w_i$  is the inverse distance weight of the  $i^{th}$  node w.r.t. point  $(x,y)$ ,  $t_i$  is the thickness of the  $i^{th}$  node, and  $N_n$  is the whole set of nodes. Shepard interpolants [95] are used to approximate the thickness inside the element space, understanding that element shape functions might lead to negative thickness values for higher-order elements [43–44]. The most important aspect comes when estimating the rate of change of compliance w.r.t. to the nodal point thicknesses. Here, the rate of change of compliance is derived w.r.t. to the TF as follows,

$$\forall i \in N_n : \partial C / \partial f_i = u^T \partial K / \partial f_i u$$

$$\partial K / \partial f_i = \sum_{j \in N_e} \partial K_{ej} / \partial f_i \quad (13)$$

$\partial C / \partial f_i$  and  $\partial K / \partial f_i$  are the rate of change of compliance and stiffness w.r.t. TF at  $i^{th}$  node that belongs to a set of nodes ( $N_n$ ). The rate of change of global stiffness ( $K$ ) is dependent on element stiffness as shown in Eq. (13) and is evaluated as follows,

The elemental stiffness matrix in 2.5D is given by.

$$\forall j \in N_e : K_{ej} = \iiint_{\Omega_{ej}} (B(x,y)^T DB(x,y)) dx dy dz = \int g(x_{\Omega_{ej}}, y_{\Omega_{ej}}) dz \quad (14)$$

Assuming that the summation of change of thickness within the element space is equal to the change of thickness at the element center and substituting thickness at element center from Eq. (12) in Eq. (14),

$$\int g(x_{\Omega_{ej}}, y_{\Omega_{ej}}) dz = g(x_{\Omega_{ej}}, y_{\Omega_{ej}}) \left( \sum_{i \in N_n, n_e} w_{ij} t_i \right) \quad (15)$$

Writing Eq. (10) in nodal thicknesses form and substituting in Eq. (15).

$$\forall i \in N_n : t_i = [f_{min} + f_i^p (f_{max} - f_{min})] b$$

$$\forall i \in (N_n, n_e), j \in N_e : \partial K_{ej} / \partial f_i = g(x_{\Omega_{ej}}, y_{\Omega_{ej}}) w_{ij} p f_i^{p-1} (f_{max} - f_{min}) b \quad (16)$$

It is essential to compute the volumetric change along with the stiffness for sensitivity analysis. As mentioned earlier, the primary reason to penalize the stiffness values is to avoid the intermediate

design values. Simultaneously, the volumetric rate is evaluated based on absolute values (i.e., Eq. (10) with no penalty) such that design values having lower stiffness occupying relatively higher volume are strategically sized down. Thus, the gradient information successfully suppresses the intermediate thickness values. So, the rate of volume change at the nodal level is evaluated similarly to the stiffness.

Element volume for the varying thickness is given by.

$$\forall j \in N_e : V_{ej} = \iiint_{\Omega_{ej}} dx dy dz = \int A(x_{\Omega_{ej}}, y_{\Omega_{ej}}) dz \quad (17)$$

Referring back to the assumption in Eq. (15),

$$\int A(x_{\Omega_{ej}}, y_{\Omega_{ej}}) dz = A(x_{\Omega_{ej}}, y_{\Omega_{ej}}) \left( \sum_{i \in N_n, n_e} w_{ij} t_i \right) \quad (18)$$

Writing the nodal thickness from Eq. (10) without penalty and substituting it in Eq. (18) to obtain the rate of change of volume w.r.t. thickness.

$$\forall i \in N_n : t_i = [f_{min} + f_i (f_{max} - f_{min})] b$$

$$\forall i \in N_n : \partial V / \partial f_i = \sum_{j \in N_e} \partial V_{ej} / \partial f_i$$

$$\forall i \in (N_n, n_e), j \in N_e : \partial V_{ej} / \partial f_i = A(x_{\Omega_{ej}}, y_{\Omega_{ej}}) w_{ij} (f_{max} - f_{min}) b \quad (19)$$

Where  $t_i$  is the thickness of the  $i^{th}$  node,  $f_i$  is the TF of the  $i^{th}$  node,  $g(x_{\Omega_{ej}}, y_{\Omega_{ej}})$  is the stiffness matrix of the  $j^{th}$  element ( $\Omega_{ej}$ ) with unit thickness,  $n_e$  is the set of nodes of the  $j^{th}$  element,  $w_{ij}$  is the inverse distance weight of  $i^{th}$  node from the center of the  $j^{th}$  element,  $A(x_{\Omega_{ej}}, y_{\Omega_{ej}})$  is the area of the  $j^{th}$  element. It should be noted that SIMTP penalizes the nodal design variable (i.e., TF). In contrast, other TO methods penalize element densities regardless of the design variable, including NDV methods.

### 3.3. Energy regularization

Jacobian is the epicenter for most mesh-related instabilities in the regular FEM apart from the numerical instabilities of the traditional TO. A primary advantage of the 2.5D SIMTP is unifying all possible instabilities to the 3D Jacobian except that caused by material non-linearity. 2.5D element presented in Section 2.1 is seemingly stable and accurate until an unusual thickness gradient (out-of-plane distortion) appears in an element, causing the Jacobian illness (mapping from global coordinates to local coordinates). A Jacobian repair method is necessary to have a stable analysis in this scenario. Adjusting nodal positions/mesh refinement through Jacobian ratios are general practices in the literature to deal with the heavily distorted elements [96]. Thus, the energy of an element

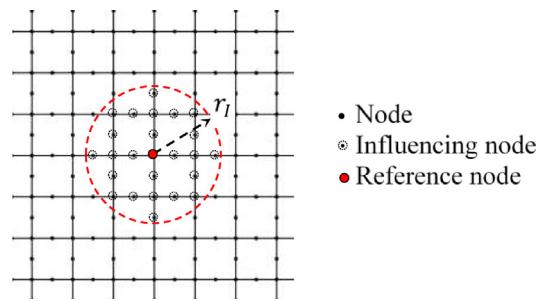


Fig. 8. Energy regularization.

is effectively maximized or relaxed. Precisely, regularization techniques [18–25] in the TO work similarly to avoid unwanted material placement. Therefore, the present study uses a general TO filter [20–21] as an energy regulator and its implementation at the nodal level is presented in Eq. (20–22). Fig. 8 illustrates the adopted filtering scheme, and Eq. (20–22) extends the derivatives in Eqs. (16) and (19) based on the filtered/unfiltered TF.

$$\begin{aligned} \forall i \in N_n : f_i &= \sum_{k \in N_{l,i}} \hat{f}_k \phi_{i,k} / \sum_{k \in N_{l,i}} \phi_{i,k} \\ \forall k \in N_{l,i} : \phi_{i,k} &= (r_{l,i} - r_{i,k}) / r_{l,i} \\ \forall k \notin N_{l,i} : \phi_{i,k} &= 0 \end{aligned} \quad (20)$$

Filtered TF's from Eq. (20) influences the sensitivity analysis (refer to Eq's. (16) and (19)) accordingly.

$$\begin{aligned} \forall i \in (N_n, n_{e,j}), j \in N_e, k \in N_{l,i} : \partial K_{ej} / \partial \hat{f}_k &= (\partial K_{ej} / \partial f_i) \partial f_i / \partial \hat{f}_k \\ \forall i \in (N_n, n_{e,j}), j \in N_e, k \in N_{l,i} : \partial V_{ej} / \partial \hat{f}_k &= (\partial V_{ej} / \partial f_i) \partial f_i / \partial \hat{f}_k \\ \forall i \in N_n, k \in N_{l,i} : \partial f_i / \partial \hat{f}_k &= \phi_{i,k} / \sum_{j \in N_{l,i}} \phi_{i,j} \end{aligned} \quad (21)$$

For un-filtered thickness.

$$\begin{aligned} \forall i \in N_n, k \in N_{l,i}, i = k : \partial f_i / \partial \hat{f}_k &= 1 \\ \forall i \in N_n, k \in N_{l,i}, i \neq k : \partial f_i / \partial \hat{f}_k &= 0 \end{aligned} \quad (22)$$

Where  $r_{l,i}$  is the influence radius of the  $i^{th}$  node,  $r_{i,k}$  is the distance of the  $k^{th}$  node from the  $i^{th}$  node,  $\phi_{i,k}$  is the influence of the  $k^{th}$  node on the  $i^{th}$  node,  $\hat{f}_k$  is the actual TF of the  $k^{th}$  node,  $N_{l,i}$  is the set of influencing nodes of the  $i^{th}$  node,  $\partial K_{ej} / \partial \hat{f}_k$ ,  $\partial V_{ej} / \partial \hat{f}_k$ ,  $\partial f_i / \partial \hat{f}_k$  is the rate of change of  $j^{th}$  element stiffness,  $j^{th}$  element volume, and scaled TF of  $i^{th}$  node, w.r.t. actual TF of  $k^{th}$  node, respectively.

#### 4. SIMTP Results, discussion and advantages

##### 4.1. 2.5D SIMTP Vs 2D SIMP

This section compares results obtained from (a) SIMP using 2D plane stress element and (b) SIMTP using a 2.5D plane stress element. As illustrated in Fig. 9, three beam design spaces were optimized using the two optimization methods for ten case scenarios, as listed in Table 2. All the models were optimized to 50% material occupancy, and analyses were performed on the Intel Xeon W-2155 processor 3.3 GHz with 64 GB RAM. The optimality criteria (OC) method, as discussed in Andreassen et al. [97], was used to identify the design variables satisfying the volumetric constraint. Results in this section were compiled using a 211-line MATLAB code modified from top88 [97]. The modified code, top211, includes (a) elements: 2D eight-noded element and 2.5D element with two-point integration rule and (b) TO methods: SIMP and SIMTP. Also, a copy of the code mentioned above and examples is presented in Appendix B for educational purposes.

An initial threshold of 5% was set for SIMTP models (see line 79 of top211) to avoid the negative Jacobian (see 184 – 210 lines of top211) that may be caused due to the high thickness gradient in an element. The above initial threshold was determined by a parametric study through varying TF at a single node while considering unit TF values at the rest of the nodes in an element such that Jacobian remains positive. Even though 2D SIMP is independent of Jacobian, SIMP models relatively have a low initial threshold to avoid singular matrices. The initial threshold in further is discussed in 4.2. The nomenclature used in the present and following sections:  $O_X$  and  $t_X$  are compliance and computational time of  $X$ ,  $e_z$  is the element size,  $r_z$  is the ratio of the radius (of influence) to  $e_z$ ,  $\Delta p$  is the penalty increment per iteration. The computational time axis in the plots is represented in the logarithmic scale.

Fig. 10 presents the optimized profiles of cantilever beam cases C1 – C4, after 150 iterations. Despite using the same parameters, optimized profiles in Fig. 10(a) and 10(b) show distinct shape out-

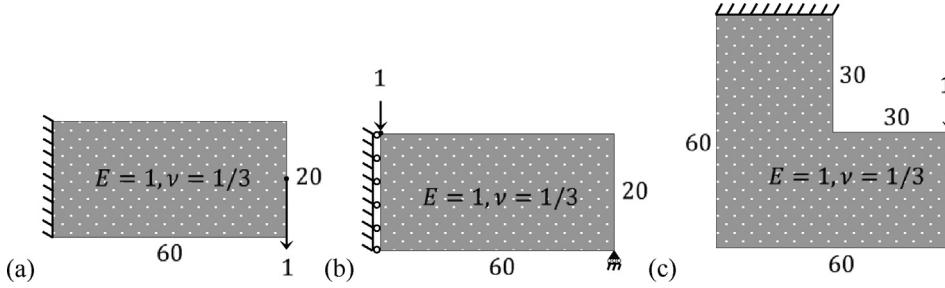


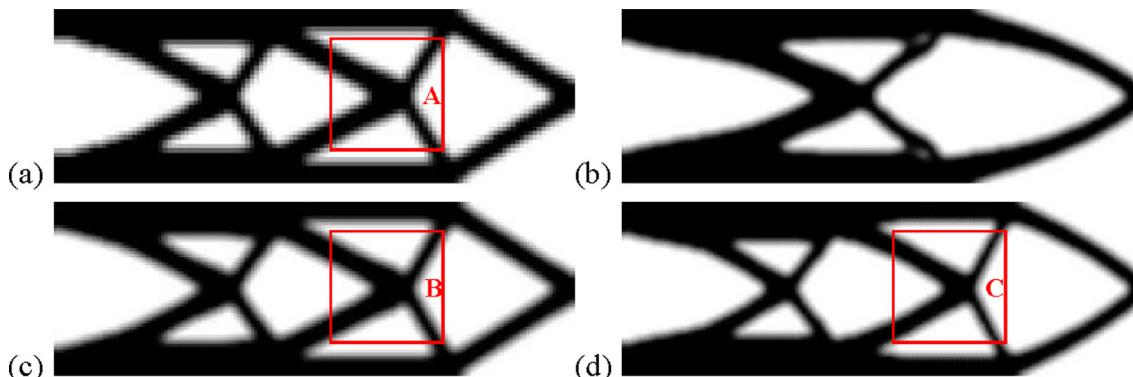
Fig. 9. Design space of the beams. (a) Cantilever; (b) MBB and (c) L-beam.

Table 2

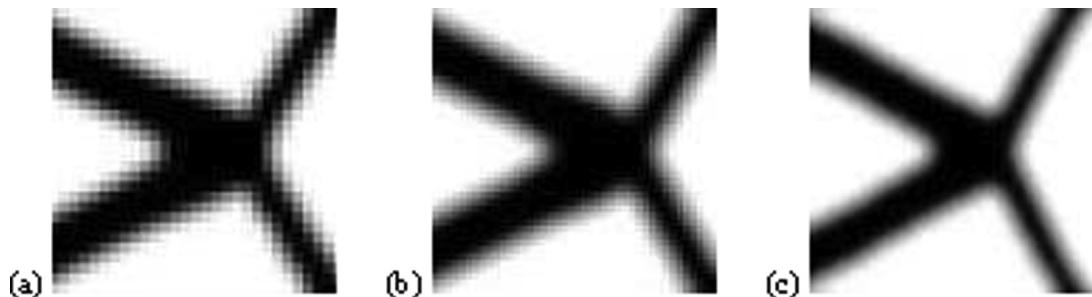
Case studies, parameters, and outcomes.

| Case | Beam Model | Optimization Method | $e_z$ | $r_z$ | Initial Threshold (%) | Penalty |      |            | Compliance | Computation time (Seconds) |
|------|------------|---------------------|-------|-------|-----------------------|---------|------|------------|------------|----------------------------|
|      |            |                     |       |       |                       | Min.    | Max. | $\Delta p$ |            |                            |
| C1   | Cantilever | SIMP                | 0.5   | 1.5   | $10^{-7}$             | 3       | 3    | 0          | 190        | 41                         |
| C2   | Cantilever | SIMTP               | 0.5   | 1.5   | 5                     | 3       | 3    | 0          | 183        | 72                         |
| C3   | Cantilever | SIMP                | 0.33  | 2.5   | $10^{-7}$             | 3       | 3    | 0          | 190        | 88                         |
| C4   | Cantilever | SIMTP               | 0.5   | 1.5   | 5                     | 1       | 3    | 0.05       | 179        | 72                         |
| C5   | MBB        | SIMP                | 0.5   | 1.5   | $10^{-7}$             | 3       | 3    | 0          | 213        | 42                         |
| C6   | MBB        | SIMTP               | 0.5   | 1.5   | 5                     | 3       | 3    | 0          | 204        | 72                         |
| C7   | L-beam     | SIMP                | 0.5   | 1.5   | $10^{-7}$             | 3       | 3    | 0          | 89         | 88                         |
| C8   | L-beam     | SIMTP               | 0.5   | 1.5   | 5                     | 3       | 3    | 0          | 88         | 177                        |
| C9   | MBB        | SIMTP               | 0.5   | 1.5   | 5                     | 1       | 3    | 0.05       | 197        | 70                         |
| C10  | L-beam     | SIMTP               | 0.5   | 1.5   | 5                     | 1       | 3    | 0.05       | 86         | 179                        |

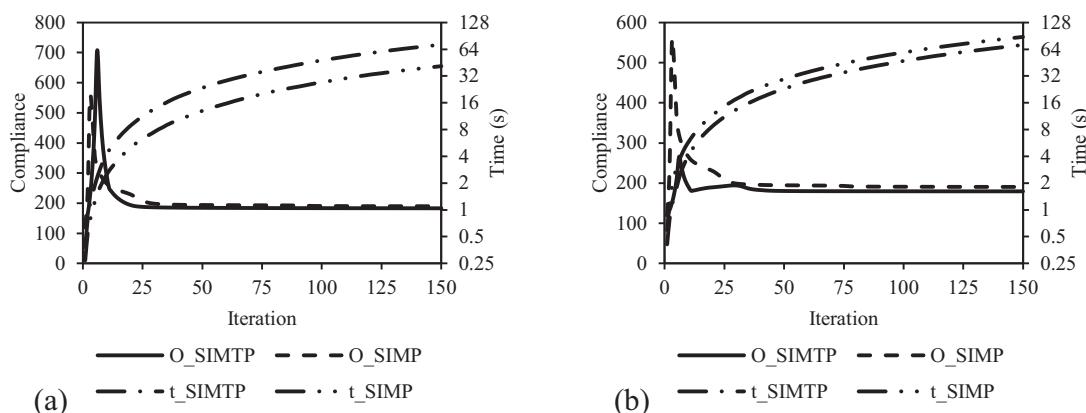
comes. Therefore, C4 was studied to achieve a similar shape outcome as C1. The parameters used in C4 were identified by randomly altering the values from C2. Likewise, C3 was studied to replicate the resolution of the C4 by changing the parameters from C1 accordingly. Noting the computational times from [Table 2](#), cantilever beam C1 consumed less computational time than its alternative C2, C3 and C4. However, it is observed from [Fig. 11](#) that C1 produced a pixelated image while C3 and C4 had better resolution, whereas C3 consumed computational time higher than C4 (also see [Fig. 12\(b\)](#)). Former observation shows that the 2.5D SIMTP is more efficient and economical in producing high-resolution images with less computational effort, while 2D SIMP compromises resolution with less computational effort or requires greater computational effort to generate high-resolution images.



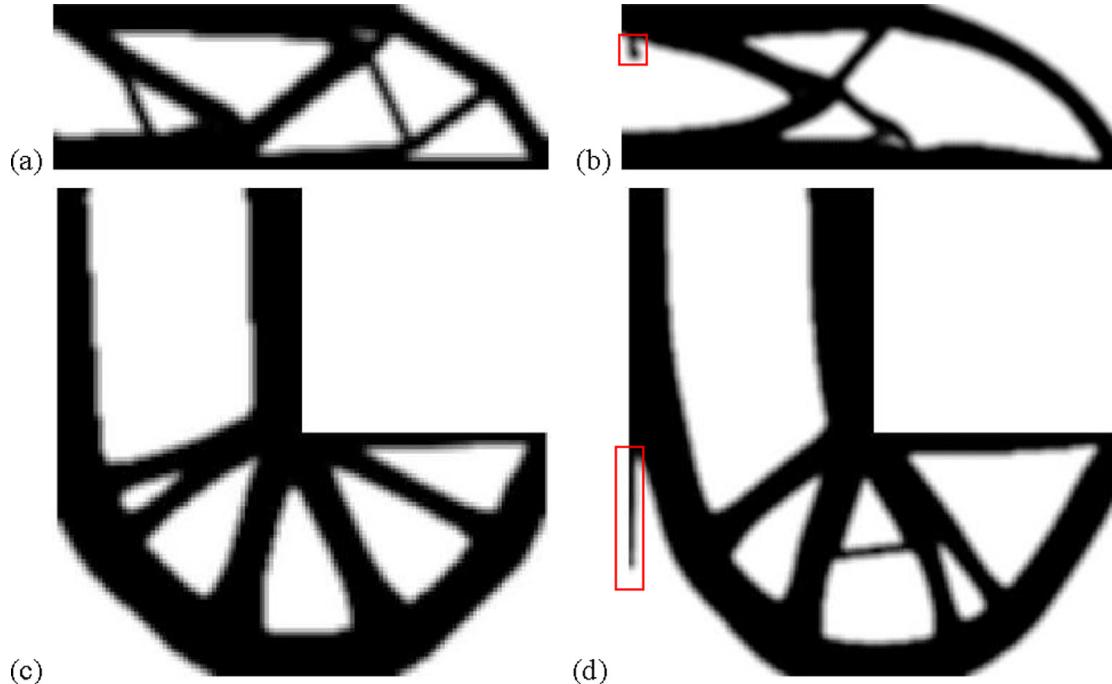
**Fig. 10.** Cantilever beam results. (a) C1; (b) C2; (c) C3 and (d) C4.



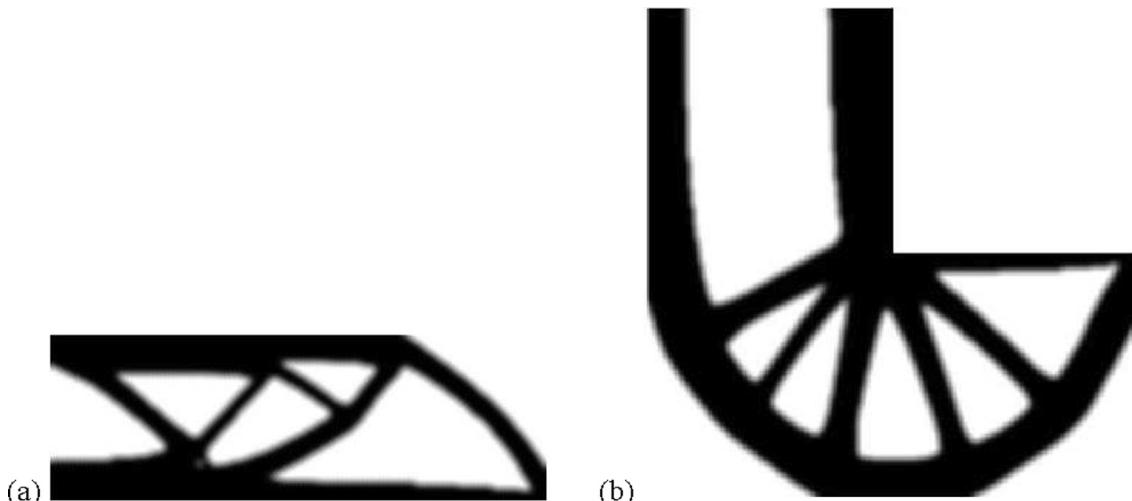
**Fig. 11.** Magnified boundaries as highlighted in [Fig. 10](#). (a) C1: region A; (b) C3: region B and (c) C4: region C.



**Fig. 12.** Performance Comparison. (a) C1 Vs C2 and (b) C3 Vs C4.



**Fig. 13.** Optimization results. (a) C5; (b) C6; (c) C7 and (d) C8.



**Fig. 14.** Effect of gradual penalty on 2.5D SIMTP. (a) C9 and (b) C10.

uniform/non-uniform discretization since it requires volume and stiffness computation for each element in either of the cases.

Optimized results of C5 – C8 after 150 iterations are presented in Fig. 13, where C6 and C8 from Fig. 13(b) and 13(d) have shown unwanted thickness distribution as highlighted. Also, such unnecessary thickness distribution disappeared with a gradual penalty increment, refer to the optimized results of C9 and C10 presented in Fig. 14. The unwanted thickness distribution in C6 and C8 may be attributed to the aggressive penalty since the higher penalty further lowers the lower TF values while values close to one remain the same. Thus, an element with an abrupt thickness gradient and an aggressive penalty causes steeper compliance rates. Also, design variables are greatly influenced by the compliance rate while unaffected by the volume rate since it remains the same throughout the optimization process (refer to line 137 of top211 code). Therefore, any adverse change in compliance rates within the element leads to the premature suppression of nodal thickness at the nodes with

relatively lower thickness, while the higher thickness at neighbouring nodes appears as the unwanted thickness distribution. So, a gradual increment between 0.05 and 0.15 is recommended for 2.5D SIMTP.

#### 4.2. 2.5D SIMTP Vs 3D SIMP

Results discussed earlier in 4.1 are based on unit thickness since 2.5D results are compared with 2D. Density as a design variable in 2D can only produce pixels irrespective of the element/nodal variable. Also, 2.5D SIMTP results presented in 4.1 showed high-resolution optimized profiles, but it is associated with the nodal design variables [31–32,43–44]. Regardless, pixels are the limited information for manufacturing optimized structural components. Therefore, 3D optimization is necessary to generate either voxels or STL.

Further, this section dedicates to exploring further merits of the 2.5D SIMTP. Firstly, a 3D design space for optimizing the MBB

**Table 3**

Model discretization, parameters, and outcomes.

| Model | Opt. Method | $e_z$ (cm)                                  | Nodes   | Total Dof's | $r_z$ | Threshold (%) |       | Penalty |      |            | Comp. (N-cm) | Com. Time (s) |
|-------|-------------|---|---------|-------------|-------|---------------|-------|---------|------|------------|--------------|---------------|
|       |             |   |         |             |       | Init.         | Disp. | Min.    | Max. | $\Delta p$ |              |               |
| M0    | SIMTP       | 0.1   | 18,381  | 36,762      | 1.5   | 5             | 1     | 1       | 3    | 0.05       | 5.91         | 37            |
| M1    | SIMP        | 0.19 in XY direction and 2.5 in Z direction | 12,116  | 36,348      | 1.5   | $10^{-7}$     | 50    | 3       | 3    | 0          | 7.04         | 56            |
| M2    | SIMP        | 0.3825 in XYZ directions                    | 13,232  | 39,696      | 1.5   | $10^{-7}$     | 50    | 3       | 3    | 0          | 7.33         | 99            |
| M3    | SIMP        | 0.1 in XY direction and 2.5 in Z direction  | 42,953  | 128,859     | 1.5   | $10^{-7}$     | 50    | 3       | 3    | 0          | 6.49         | NA            |
| M4    | SIMP        | 0.1 in XYZ directions                       | 632,681 | 1,898,043   | 1.5   | $10^{-7}$     | 50    | 3       | 3    | 0          | 5.68         | NA            |

Note: Opt. – optimization; Init. – Initial; Disp. – Display; Min. – Minimum; Max. – Maximum; Comp. – Compliance; Com. – Computation

beam using 2.5D SIMTP and 3D SIMP is illustrated in Fig. 16 (a). Optimization was dealt with within the elastic range of ABS M30i material, and its properties are presented in Fig. 16(b). Half of the beam was only modelled in 2.5D and 3D, taking advantage of symmetry.

Table 3 presents the details, parameters and outcomes of the various models. Four types of discretizations were chosen for the 3D model, (a) M1: matching the total number of degrees of freedom with M0 but having a single element in the thickness direction; (b) M2: matching the total number of degrees of freedom with M0; (c) M3: matching the element size with M0 but having single element in the thickness direction and (d) M4: matching the element size with M0, also refer to Table 3. M1 and M3 were discretized with a single element in the thickness direction since M0 (2.5D SIMTP) assumes a single element in the respective direction. Twenty-noded serendipity elements were used to discretize 3D models. All 3D models were optimized using a 145-line MATLAB code, topMBB3D20N limited to MBB beam. The 145-line code presented in Appendix C is exploited from top3d [98], and it is worth noting that lines 102 – 109 of the modified code introduce an iterative procedure since M4 has many nodes that require

substantial computational space and efforts for direct matrix inversion. Lines of top211, as shown in Fig. 15 (a), were modified to lines in Fig. 15(b) to analyse the M0 model, where  $th$  is the model's thickness, 2.5 cm in this case.

Further, a display threshold for densities/TF's is applied, as summarized in Table 3. The display threshold in 3D SIMP models is necessary since the design variable is material dependent, and describing the material occupancy, especially intermediate densities in 3D space, is highly complicated. Therefore, elements with densities lower than a set threshold are omitted from the display and the rest of the elements are set to unit densities for generating boundary surfaces. Further, a smoothing function must construct the surface before converting it to an STL. Elements having less than 50% density were removed from the display of 3D SIMP models, and the value is used as suggested in top3d [98]. This display threshold is far higher than the combined initial and display thresholds of 2.5D SIMTP. Unlike SIMP, SIMTP directly produces a boundary surface, refer to Fig. 7.

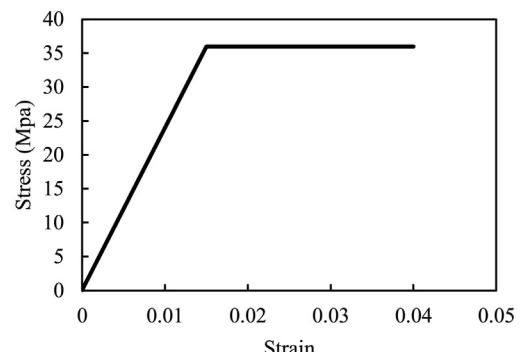
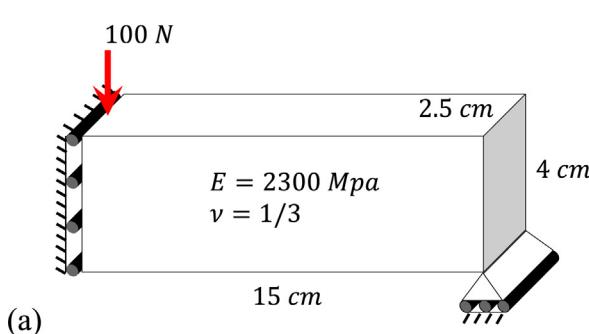
A boundary surface of optimized M0 after removing faces below a threshold TF is presented in Fig. 17, and this surface is later extended at the design space boundaries to form an entire bound-

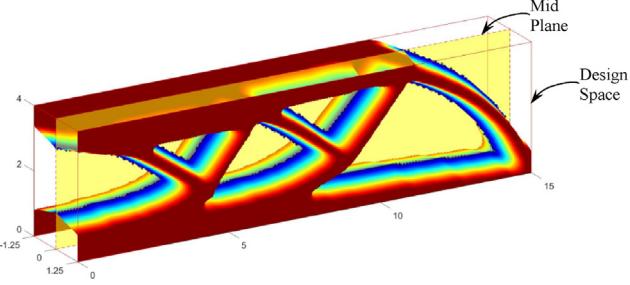
```

18   V0=a*b;
43   V0=(a*b-c*d);
135  else, dch=0.001; dv1=repmat(ez*ez*w*(E0-Em),et,1); dv=L(f(dv1'));
(a) 153 C=U'*K*U;dt=pn*(E0-Em)*xs(ep).^(pn-1).*w;
186 tp=(Em+xs(ep).^pn*(E0-Em));
199 tn=Em+xs(ep)*(E0-Em);

18   V0=a*b*th;
43   V0=(a*b-c*d)*th;
135  else, dch=0.001; dv1=repmat(ez*ez*w*(E0-Em)*th,et,1); dv=L(f(dv1'));
153 C=U'*K*U;dt=pn*(E0-Em)*xs(ep).^(pn-1).*w*th;
186 tp=(Em+xs(ep).^pn*(E0-Em))*th;
199 tn=(Em+xs(ep)*(E0-Em))*th;

```

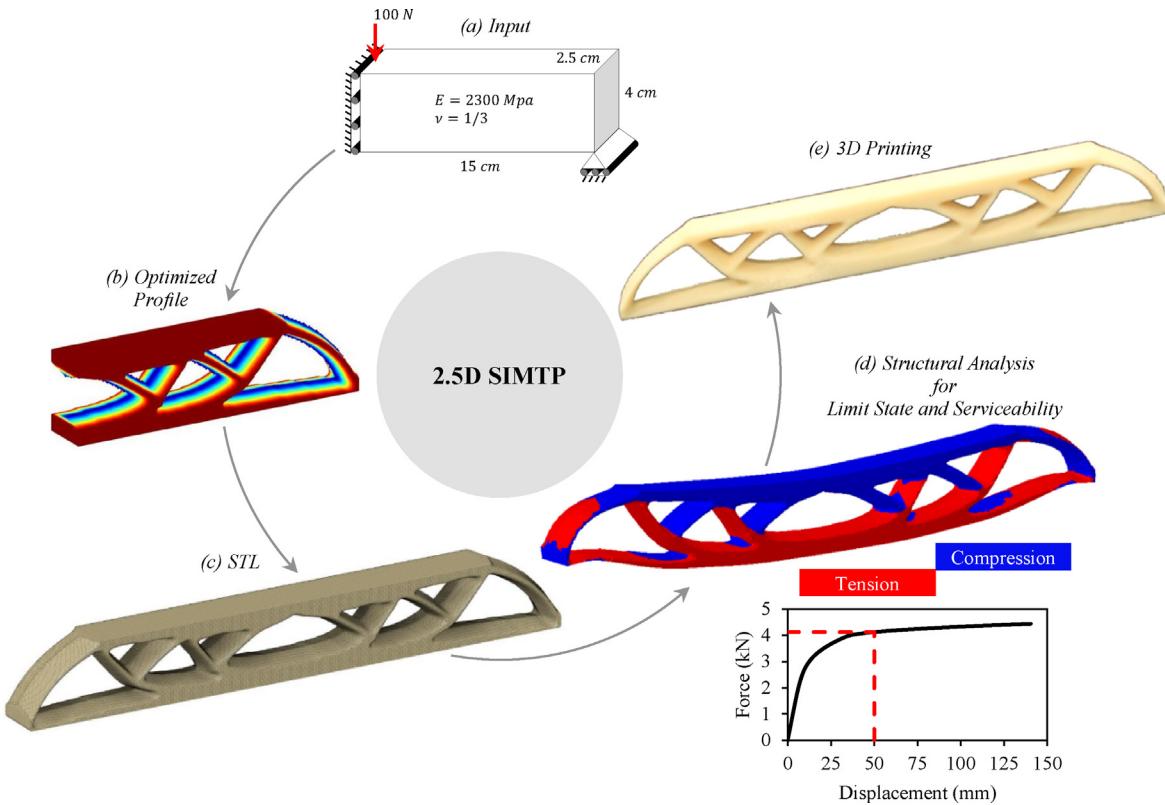
**Fig. 15.** Matlab modifications for 2.5D analysis. (a) Unit thickness and (b) Absolute thickness.**Fig. 16.** (a) 3D design space for optimizing an MBB beam and (b) Stress-strain curve of ABS M30i.



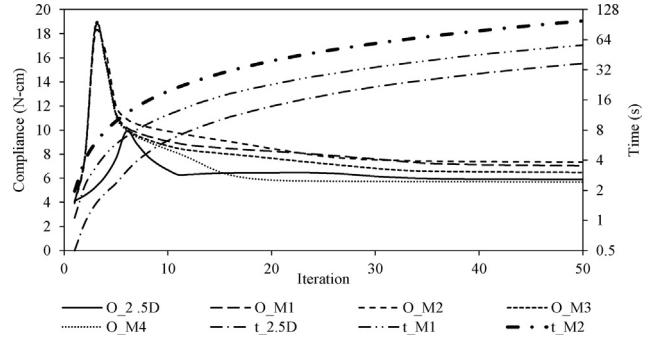
**Fig. 17.** Boundary surface of M0 model.

ing surface, as shown in Fig. 18(b). Further, the above surface can be mirrored at the mid-span to form the full-scale MBB beam and converted to STL for 3D printing (see Fig. 18(c)). The STL can then be further processed to limit-state and serviceability check for manufacturing/3D printing. An open-source MATLAB code, stl-write [99], converted the M0's resulting surface to STL. A solid model converted from STL was exported for finite element analysis using the material properties (see Fig. 16), and stress output at 50 mm mid-span deflection is presented in Fig. 18(d). A 3D printed prototype at PolyU's U3DP laboratory using ABS M30i material is also presented in Fig. 18(e) for display purposes.

Further, Fig. 19 and Table 3 compare compliance and performance of all models, where computational time for M0, M1 and M2 are only compared since it is inappropriate to compare M3 and M4 for their relatively large number of degrees of freedom. Observations have shown that M1 and M2 required computational efforts higher than M0. Optimized profiles of all models after 50 iterations are presented in Fig. 20. Optimized profiles show that the 2.5D SIMTP can produce high-resolution smooth boundaries with less computational effort relative to 3D. Despite the mesh quantity, M1 and M3 optimized profiles have better shape outcomes than M2 and M4.



**Fig. 18.** 2.5D SIMTP's ecosystem.

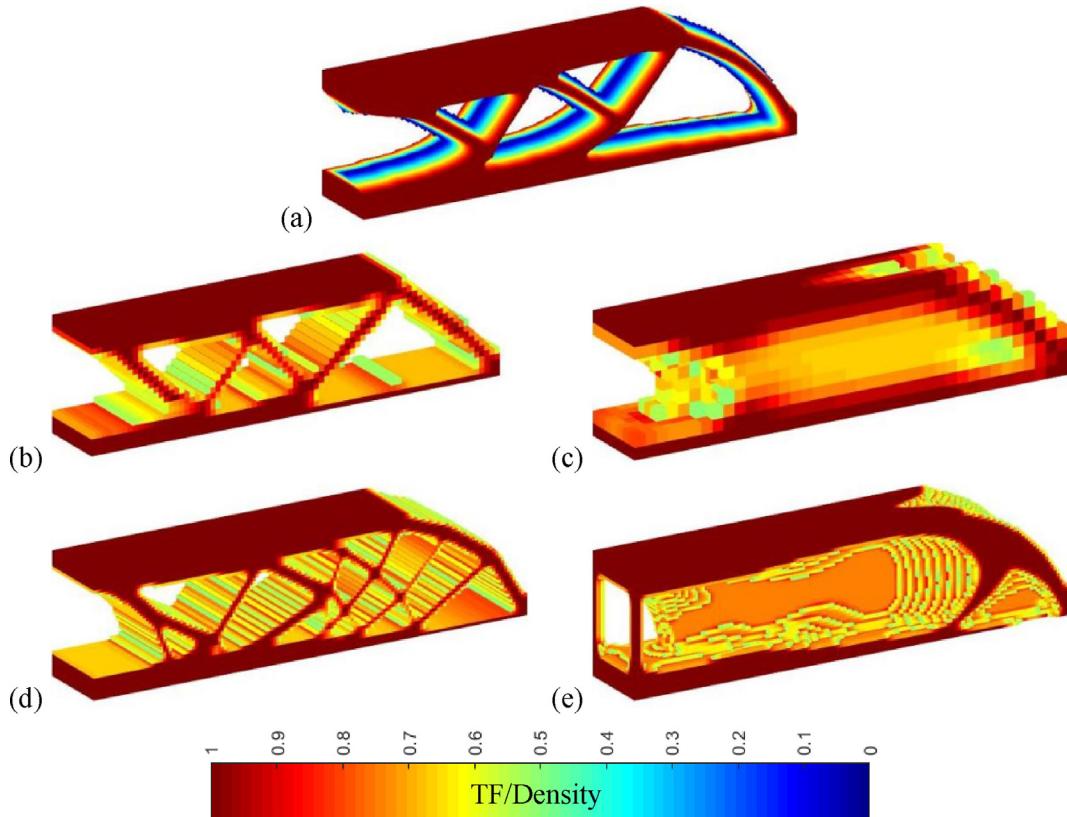


**Fig. 19.** Performance comparison of 2.5D SIMTP and 3D SIMP.

## 5. Concluding remarks

SIMTP and 2.5D element proposed in this paper are a progression of SIMP, and the findings are preliminary. SIMTP presents thickness as a design variable where a 2.5D element connects SIMTP with FEM. The 2.5D element assumes the state of stress as 2D and volume as 3D. Also, the developed element is limited to in-plane loading systems and sensitive to extreme thickness variations. It is suggested to use a sufficient number of 2.5D elements to yield computationally efficient results.

Stable Jacobian in 2.5D SIMTP ensures avoiding general TO instabilities. However, unwanted thickness distribution appears when the aggressive penalty is imposed, and it is recommended to use a gradual increment of the penalty between 0.05 and 0.15. SIMTP uses a higher initial threshold relative to regular TO methods to maintain a positive Jacobian in the 2.5D element. Nevertheless, traditional 3D optimization methods use higher display thresholds for boundary extraction. In addition, they require a smoothing algorithm for surface construction, while SIMTP is com-



**Fig. 20.** Optimized profiles. (a) M0; (b) M1; (c) M2; (d) M3 and (e) M4.

putationally economical, efficient, and straightforward in yielding high-resolution boundary surfaces.

2D optimization methods require less computational effort than 2.5D SIMTP while using the same and uniform discretization. Also, present TO methods compromise resolution with less computational effort or require greater computational effort to generate high-resolution profiles. Besides, it is easy to implement material and geometric nonlinearity in 2.5D SIMTP since the constitutive matrix is free from design variables.

The present study uses traditional filtering schemes to maintain the stability of Jacobian. Furthermore, 2.5D SIMTP behavior needs to be studied using mesh refinement with actual/unfiltered thickness distribution.

#### Declaration of Competing Interest

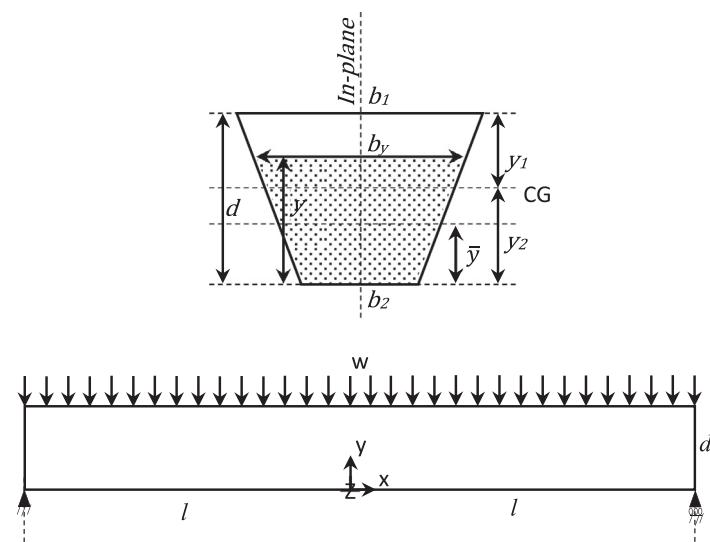
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Acknowledgment

The authors would like to acknowledge the Hong Kong Polytechnic University funding for this work through a PhD scholarship to Tejeswar Yarlagadda and Zixin Zhang. The University Research Facility in 3D Printing (U3DP) laboratory at the Hong Kong Polytechnic University provided the 3D printing facilities to print the prototype. Also, the authors would like to thank the authors of “top88” from the Department of Solid Mechanics, the Technical University of Denmark, for providing the open-source code for education purposes.

#### Appendix A

In general, analytical solutions for 2D elasticity problems are derived based on the Airy's stress function, and these solutions are limited to a constant width of a beam. However, case-I has linearly varying width across the span (trapezoidal beam), where general Airy's stress function does not hold good. The analytical solution presented here is inspired by Airy's stress function. The derived solution is limited to a simply supported trapezoidal beam assuming plane stress conditions.



Stress equilibrium equations for a trapezoidal beam.

$$\frac{\partial(\sigma_{xx}b_y)}{\partial x} + \frac{\partial(\sigma_{xy}b_y)}{\partial y} = 0; \frac{\partial(\sigma_{xy}b_y)}{\partial x} + \frac{\partial(\sigma_{yy}b_y)}{\partial y} = 0$$

Stresses should be of the following form in order to obey the above equilibrium equations.

$$\sigma_{xx} = \frac{1}{b_y} \frac{\partial^2 \phi}{\partial y^2}; \sigma_{yy} = \frac{1}{b_y} \frac{\partial^2 \phi}{\partial y^2}; \sigma_{xy} = -\frac{1}{b_y} \frac{\partial^2 \phi}{\partial x \partial y}$$

Stress, strain and displacement relationship (Poisson ratio is zero).

$$\epsilon_{xx} = \frac{\partial u_x}{\partial x} = \frac{\sigma_{xx}}{E}; \epsilon_{yy} = \frac{\partial u_y}{\partial y} = \frac{\sigma_{yy}}{E}; \epsilon_{xy} = \frac{1}{2} \left( \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right) = \frac{\sigma_{xy}}{E}$$

Since strains are interdependent, they should satisfy the following compatibility equation.

$$\frac{\partial^2 \epsilon_{xx}}{\partial y^2} + \frac{\partial^2 \epsilon_{yy}}{\partial x^2} = 2 \frac{\partial^2 \epsilon_{xy}}{\partial x \partial y}$$

Now substituting stress-strain relations and earlier assumed stress forms in the above compatibility equation,  $\phi$  should satisfy the following criteria.

$$b_y^2 \left( \frac{\partial^4 \phi}{\partial y^4} + \frac{\partial^4 \phi}{\partial x^4} + 2 \frac{\partial^4 \phi}{\partial x^2 \partial y^2} \right) - 2k_0 b_y \left( \frac{\partial^3 \phi}{\partial y^3} + \frac{\partial^3 \phi}{\partial x^2 \partial y} \right) + 2k_0^2 \frac{\partial^2 \phi}{\partial y^2} = 0$$

Any general function can satisfy the above equation, for example,  $\phi = 0$ , but it has to satisfy the boundary conditions of case-I, which are as follows.

|  |   |
|--|---|
| Shear Stress at the top and bottom of the beam | : $\sigma_{xy}(x, 0) = 0; \sigma_{xy}(x, d) = 0$      |
| Vertical Stress at the bottom of the beam      | : $\sigma_{yy}(x, 0) = 0$                             |
| Vertical Stress at the top of the beam         | : $\sigma_{yy}(x, d) = -\frac{w}{b_1}$                |
| Axial force at ends                            | : $\int_{-d}^d \sigma_{xx}(\pm l, y) b_y dy = 0$      |
| Shear forces at ends                           | : $\int_{-d}^d \sigma_{xy}(\pm l, y) b_y dy = \pm wl$ |
| Moment at ends                                 | : $\int_{-d}^d \sigma_{xx}(\pm l, y) b_y y dy = 0$    |
| Vertical deformation at both ends              | : $u_y(\pm l, 0) = 0$                                 |
| Axial deformation at the left end              | : $u_x(-l, 0) = 0$                                    |

Assuming second to sixth-degree power series as a stress function since (1) criterion equation for  $\phi$  is a fourth-order derivative; (2) width is a dependent variable; (3) stress components are second-order derivatives.

$$\phi = \sum_{i=0}^6 \sum_{j=0}^6 \delta_{ij} C_{ij} x^i y^j$$

$$\delta_{ij} = \begin{cases} 1, & \text{if } 2 \leq i+j \leq 6 \\ 0, & \text{otherwise} \end{cases}$$

Considering the x-symmetry of the problem,  $\phi$  should be an even function. Therefore, coefficients corresponding to the odd-order should be zero.

$$\forall i \in [1, 3, 5], \forall j \in Z_{[0,6]}, 2 \leq i+j \leq 6 : C_{ij} = 0$$

Now solving the above equations for  $\phi$  satisfying the earlier stated criterion and boundary conditions.

$$\phi = \sum_{j=2}^6 (C_{0j} y^j) + \sum_{j=2}^4 (C_{2j} x^2 y^j)$$

$$\sigma_{xx} = \frac{1}{b_y} \left( 2C_{02} + 6C_{03}y + 12C_{04}y^2 + 20C_{05}y^3 + 30C_{06}y^4 + 2C_{22}x^2 + 6C_{23}x^2y + 12C_{24}x^2y^2 \right)$$

$$\sigma_{yy} = \frac{1}{b_y} (2C_{22}y^2 + 2C_{23}y^3 + 2C_{24}y^4); \sigma_{xy} = -\frac{1}{b_y} (4C_{22}xy + 6C_{23}xy^2 + 8C_{24}xy^3)$$

$$u_x = \frac{1}{Eb_y} \left( 2C_{02}x + 6C_{03}xy + 12C_{04}xy^2 + 20C_{05}xy^3 + 30C_{06}xy^4 + \frac{2}{3}C_{22}x^3 + 2C_{23}x^3y + 4C_{24}x^3y^2 + a_1 b_y \right)$$

$$u_y = \frac{1}{Eb_y} \left( \frac{2}{3}C_{22}y^3 + \frac{1}{2}C_{23}y^4 + \frac{2}{5}C_{24}y^5 + g(x, y) \right)$$

$$\begin{bmatrix} 4k_0^2 & -12b_2 k_0 & 24b_2^2 & 0 & 0 & 8b_2^2 & 0 & 0 \\ 0 & 0 & 0 & 120b_2^2 & 0 & 8b_2 k_0 & 24b_2^2 & 0 \\ 0 & 0 & 0 & 120b_2 k_0 & 360b_2^2 & 0 & 36b_2 k_0 & 48b_2^2 \\ 0 & 0 & 0 & 0 & 0 & 4k_0^2 & -12b_2 k_0 & 24b_2^2 \\ 0 & 0 & 0 & 0 & 0 & 4d & 6d^2 & 8d^3 \\ 0 & 0 & 0 & 0 & 0 & 2d & 2d^2 & 2d^3 \\ 2 & 3d & 4d^2 & 5d^3 & 6d^4 & 2l^2 & 3l^2 d & 4l^2 d^2 \\ 1 & 2d & 3d^2 & 4d^3 & 5d^4 & l^2 & 2l^2 d & 3l^2 d^2 \end{bmatrix}$$

$$\begin{bmatrix} C_{02} \\ C_{03} \\ C_{04} \\ C_{05} \\ C_{06} \\ C_{22} \\ C_{23} \\ C_{24} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -\frac{w}{d} \\ 0 \\ 0 \end{bmatrix}$$

$$g(x, y) = \begin{cases} \frac{k_0}{b_y} \left( C_{02}x^2 + 3C_{03}x^2y + 6C_{04}x^2y^2 + 10C_{05}x^2y^3 + 15C_{06}x^2y^4 + \frac{C_{22}x^4}{6} + \frac{C_{23}x^4y}{2} + C_{24}x^4y^2 \right) \\ - \left( 4C_{22}x^2y + 6C_{23}x^2y^2 + 8C_{24}x^2y^3 + 3C_{03}x^2 + 12C_{04}x^2y \right. \\ \left. + 30C_{05}x^2y^2 + 60C_{06}x^2y^3 + \frac{C_{23}x^4}{2} + 2C_{24}x^4y + a_2 b_y \right) \end{cases}$$

$$k_0 = \frac{b_1 - b_2}{d}; a_1 = \frac{(2C_{02}l + \frac{2}{3}C_{22}l^3)}{b_2}; \\ a_2 = \frac{k_0(C_{02}l^2 + \frac{1}{6}C_{22}l^4) - b_2(3C_{03}l^2 + \frac{1}{2}C_{23}l^4)}{b_2^2}$$

For example, substituting the beam properties in  $u_y$ .

$$b_1 = 0.1m; b_2 = 0.05m; d = 0.1m; l = 0.5m; E = 2500 \frac{N}{m^2};$$

$$S = 0.1 \frac{N}{m^2}; w = b_1 S$$

$$u_y(0, 0) = 8.804mm$$

**Appendix B**

Examples:

```
C1 : top211([60,20],0.5,-1,'mid',[1,1/3],'CB','SIMP',0.5,3,1.5,2)
C2 : top211([60,20],0.5,-1,'mid',[1,1/3],'CB','SIMTP',0.5,3,1.5,2)
C3 : top211([60,20],1/3,-1,'mid',[1,1/3],'CB','SIMP',0.5,3,2.5,2)
C4 : top211([60,20],0.5,-1,'mid',[1,1/3],'CB','SIMTP',0.5,[1,0.05,3],1.5,2)
C5 : top211([60,20],0.5,-1,'top',[1,1/3],'MBB','SIMP',0.5,3,1.5,2)
C6 : top211([60,20],0.5,-1,'top',[1,1/3],'MBB','SIMTP',0.5,3,1.5,2)
C7 : top211([60,60,30,30],0.5,-1,'top',[1,1/3],'LB','SIMP',0.5,3,1.5,2)
C8 : top211([60,60,30,30],0.5,-1,'top',[1,1/3],'LB','SIMTP',0.5,3,1.5,2)
C9 : top211([60,20],0.5,-1,'top',[1,1/3],'MBB','SIMTP',0.5,[1,0.05,3],1.5,2)
C10: top211([60,60,30,30],0.5,-1,'top',[1,1/3],'LB','SIMTP',0.5,[1,0.05,3],1.5,2)
```

MATLAB top211 code:

```
1 %%% AN 211 LINE TOPOLOGY OPTIMIZATION CODE FOR 2.5D SIMTP May, 2021 %%%
2 function top211(domain,ez,load,loc,matprop,prob,Opt,volfrac,penal,rmin,ft)
3 % SHEPARD WEIGHTS
4 close all; w=[1/12*ones(1,4) 1/6*ones(1,4)]; f=@(x) reshape(x,[],1);
5 % PROBLEM TYPE
6 if strcmpi(prob,{ 'CB' , 'MBB' })
7     % Corner node coordinates for Cantilever beam(CB) or MBB beam
8     a=domain(1); b=domain(2); nX1=round(a/ez); nY1=round(b/ez);
9     [aX,bY]=meshgrid(0:nX1,0:nY1); ccX=aX*ez; ccY=bY*ez;
10    nc=(nX1+1)*(nY1+1); ns=reshape(1:nc,nY1+1,[]);
11    cc=zeros(nc,2); cc(ns(:,1))=[ccX(:,1) ccY(:,1)];
12    % Corner node topology
13    c1=f(ns(1:end-1,1:end-1)); c2=f(ns(1:end-1,2:end));
14    c3=f(ns(2:end,2:end)); c4=f(ns(2:end,1:end-1));
15    % Loading element
16    if strcmpi(loc,'mid')&&strcmpi(prob,'CB'), le=(nX1-1)*nY1; end
17    % Initial Volume
18    V0=a*b;
19 elseif strcmpi(prob,'LB')
20     % Corner node coordinates for L-beam
21     a=domain(1); b=domain(2); c=domain(3); d=domain(4);
22     neX=round(a/ez); neY=round(b/ez);
23     ncX=round(c*neX/a); ndY=round(d*neY/b);
24     nX1=neX-ncX; nY1=neY-ndY; nX2=ncX; nY2=nY1; nX3=nX1; nY3=ndY;
25     [X1,Y1]=meshgrid(0:nX1,0:nY1); [X2,Y2]=meshgrid(1:nX2,0:nY2);
26     [X3,Y3]=meshgrid(0:nX3,1:nY3);
27     cX1=X1*ez; cY1=Y1*ez; cX2=(a-c)+X2*ez; cY2=Y2*ez;
28     cX3=X3*ez; cY3=(b-d)+Y3*ez;
29     N1=(nX1+1)*(nY1+1); N2=nX2*(nY2+1); N3=(nX3+1)*nY3;
30     nc=N1+N2+N3; cc=zeros(nc,2);
31     n0=reshape(1:(N1+N3),nY1+nY3+1,[]); n1=n0(1:nY1+1,:);
32     n2=N1+N3+reshape(1:N2,nY2+1,[]); n3=n0(nY1+2:end,:);
33     cc(n1(:,1))=[cX1(:,1) cY1(:,1)]; cc(n2(:,1))=[cX2(:,1) cY2(:,1)];
34     cc(n3(:,1))=[cX3(:,1) cY3(:,1)];
35     % Corner node topology
36     c1=[f(n0(1:(nY1+nY3),1:end-1));n1(1:end-1,end);f(n2(1:end-1,1:end-1))];
37     c2=[f(n0(1:(nY1+nY3),2:end));f(n2(1:end-1,:))];
38     c3=[f(n0(2:end,2:end));f(n2(2:end,:))];
39     c4=[f(n0(2:end,1:end-1));n1(2:end,end);f(n2(2:end,1:end-1))];
40     % Loading element
41     if strcmpi(loc,'mid'), le=nX1*(nY1+nY3)+(nX2-1)*nY2; end
42     % Initial Volume
43     V0=(a*b-c*d);
44 end
```

```

45 % Nodal coordinates of eight noded element and element topology arrangement
46 ec=[c1 c2 c3 c4]; eds=reshape(ec(:,[1:4,2:4,1]),[],2);
47 [en,~,ix]=unique(sort(eds,2), 'rows'); em=reshape(ix,[],4)+size(cc,1);
48 cm=(cc(en(:,1),:)+cc(en(:,2),:))/2; cd=[cc;cm]; ep=[ec em];
49 %% DEFINE LOADS AND SUPPORTS
50 et=size(ep,1); nt=size(cd,1); adfs=(1:2*nt)'; U=zeros(2*nt,1);
51 if any(strcmpi(prob,{ 'CB', 'LB' }))
52     if strcmpi(loc,'top'), lid=2*nc;
53     elseif strcmpi(loc,'mid')
54         if rem(nY1,2), eid=le+round(nY1/2); lid=2*ep(eid,6);
55         else, eid=le+nY1/2; lid=2*ep(eid,3);
56     end
57     else, lid=2*(nc-nY1);
58 end
59 if strcmpi(prob,'CB'), dfs=[(1:nY1+1)'; ep(1:nY1,8)];
60 else, dfs=[(1:nX1+1)'*(nY1+nY3+1); ep((1:nX1)'*(nY1+nY3),7)];
61 end
62 fxdfs=sort([2*dfs-1;2*dfs]);
63 elseif strcmpi(prob,'MBB')
64     if strcmpi(loc,'top'), lid=2*(nY1+1);
65     elseif strcmpi(loc,'mid')
66         if rem(nY1,2), eid=round((nY1)/2); lid=2*ep(eid,8);
67         else, eid=(nY1)/2; lid=2*ep(eid,4);
68     end
69     elseif strcmpi(loc,'bot'), lid=2;
70 end
71 fxdfs=sort([(2*[(1:nY1+1)'; ep(1:nY1,8)]-1);2*(nc-nY1)]);
72 end
73 F = sparse(lid,1,load,2*nt,1); fdffs=setdiff(adfs,fxfds);
74 %% MATERIAL PROPERTIES AND CONSTITUTIVE MATRIX
75 if isempty(matprop), E=1000; nu=0.3;
76 elseif length(matprop)==1, nu=0.3;
77 else, E=matprop(1); nu=matprop(2);
78 end
79 if strcmpi(Opt,'SIMTP'), Em=0.05; M=0; else, Em=1e-9; M=1; end
80 E0=1; D=E/(1-nu^2)*[1 nu 0;nu 1 0;0 0 (1-nu)/2];
81 %% PENALTY FUNCTION
82 if length(penal)==1, pl=penal; dp=0; ph=penal;
83 elseif length(penal)==2, dp=diff(penal)/100; pl=penal(1); ph=penal(2);
84     if dp<0, disp('check penalty limits'); return; end
85 elseif length(penal)==3, pl=penal(1); dp=penal(2); ph=penal(3); ck=(ph-pl);
86     if ck<0||dp>ck||dp<0, disp('check penalty limits'); return; end
87 end
88 %% STIFFNESS MATRIX
89 % Preparing Jacobian
90 dNr = [-0.6830,-0.2277,-0.1830,-0.0610,0.9107,0.3333,0.2440,-0.3333;-0.6830,-0.0610,-0.1830,-0.2277,-0.3333, 0.2440, 0.3333, 0.9107;0.2277, 0.6830, 0.0610, 0.1830,-0.9107, 0.3333,-0.2440,-0.3333;-0.0610,-0.6830,-0.2277,-0.1830,-0.3333,-0.0610,-0.6830,-0.2277,-0.1830,-0.3333, 0.9107, 0.3333, 0.2440;0.1830, 0.0610, 0.6830, 0.2277,-0.2440, 0.3333,-0.9107,-0.3333;0.1830, 0.2277, 0.6830, 0.0610,-0.3333,-0.9107, 0.3333,-0.2440;-0.0610,-0.1830,-0.2277,-0.6830, 0.2440, 0.3333, 0.9107,-0.3333;0.2277, 0.1830,-0.6830, 0.0610,-0.3333,-0.9107, 0.3333,-0.2440;-0.0610, 0.6830,-0.3333,-0.2440, 0.3333,-0.9107];
91 JTx=dNr*cd(ep(1,:),1); JTy=dNr*cd(ep(1,:),2);
92 JT(1:4:16,:) = JTx(1:2:end,:); JT(3:4:16,:) = JTy(1:2:end,:);
93 JT(2:4:16,:) = JTx(2:2:end,:); JT(4:4:16,:) = JTy(2:2:end,:);
94 JT=reshape(JT,2,2,4); dJT=JT(1,1,:,:).*JT(2,2,:,:)-JT(1,2,:,:).*JT(2,1,:);
95 % Preparing Strain Matrix and Stiffness matrix
96 dN=zeros(2,8,4);
97 for i=1:4, ix=[2*i-1;2*i]; dN(:,:,:,i)=JT(:,:,i)\dNr(ix,:); end

```



```

108      Y(:,1:8)=eY;      Y(:,9:16)=repmat(eY(:,1:4),1,2);  Y(:,17:20)=eY(:,5:8);
109      sx=ds*X';      sy=ds*Y';
110      S(1:9:72,:)=sx(1:3:end,:);  S(2:9:72,:)=sy(2:3:end,:);
111      S(3:9:72,:)=sx(3:3:end,:);  S(4:9:72,:)=sy(1:3:end,:);
112      S(5:9:72,:)=sy(2:3:end,:);  S(6:9:72,:)=sy(3:3:end,:);
113  end
114 % PREPARE FINITE ELEMENT ANALYSIS
115 emt=zeros(et,16);  emt(:,2:2:end)=ep*2;emt(:,1:2:end)=ep*2-1;
116 I=reshape(repmat((1:16),16,1),1,[]);      J=repmat(1:16,1,16);
117 iK=f(emt(:,I)');  jK=f(emt(:,J)');
118 % PREPARE FILTER
119 if strcmpi(Opt,'SIMP')
120     h=@(x) sum(x.*[-1/4*ones(1,4) 1/2*ones(1,4)],2);
121     eXc=h(eX);  eYc=h(eY);  eC=[eXc eYc];
122     x=ones(et,1);
123 else,  eC=cd;  x=ones(nt,1);
124 end
125 rmin = rmin+1e-3;
126 [Id,r]=rangesearch(eC,eC,rmin*ez);
127 w0=cell2mat(arrayfun(@(i) [i*ones(length(Id{i}),1) Id{i}' (rmin-r{i})']...
128 ,(1:length(eC))', 'un',0));
129 H=sparse(w0(:,1),w0(:,2),w0(:,3),max(w0(:,1)),max(w0(:,1)),nzmax(w0(:,3)));
130 Hs=sum(H,2);
131 % INITIALIZE ITERATION
132 L=@(x) accumarray(f(ep'),x);    G=@(x) repmat(x,1,1,8,1);
133 xs=x;  lp=0;  lt=0;  ch=1;  maxloop=150;
134 if M,  dch=0.01;  dv=ones(et,1)*ez*ez;
135 else,  dch=0.001;  dv1=repmat(ez*ez*w*(E0-Em),et,1);  dv=L(f(dv1'));  ft=2;
136 end
137 if ft~=1, dv(:)= H*(dv(:)./Hs); end, figure()
138 % START ITERATION
139 while ch > dch && lp<maxloop
140     % PENALTY
141     tic;  lp=lp+1;  pn=pl+dp*(lp-1);  if pn>ph,  pn=ph;  end
142     % FE-ANALYSIS
143     if M,  sK=KE(:)*(Em+xs(:).^pn*(E0-Em));
144     else,  sK=sum(K1.*J3d(),3);
145     end
146     K=sparse(iK,jK,sK(:));  K=(K+K')/2;
147     U(fdः)=K(fdः,fdः)\f(fdः);
148     % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
149     if M
150         ce=sum((U(emt)*KE).*U(emt),2);  C=sum((Em+xs.^pn*(E0-Em)).*ce);
151         dc=-pn*(E0-Em)*xs.^pn.*ce;
152     else
153         C=U'*K*U;dt=pn*(E0-Em)*xs(ep).^(pn-1).*w;
154         U1=G(reshape(U(emt)',[],1,1,et));
155         U2=G(reshape(U(emt)',1,[],1,et));
156         dK=KE.*reshape(dt',1,1,8,et);
157         dc1=sum(sum(U1.*dK.*U2,1),2);  dc=-L(dc1(:));
158     end
159     % FILTERING/MODIFICATION OF SENSITIVITIES
160     if ft==1&&M,  dc(:)= H*(x(:).*dc(:))./Hs./max(1e-3,x(:));
161     else,  dc(:)=H*(dc(:)./Hs);
162     end
163     % OPTIMALITY CRITERIA UPDATE OF DESIGN VARIABLES
164     l1=0;  l2=1e9; if M,  mv=0.2; else,  mv=0.1; end
165     while (l2-l1)/(l1+l2)>1e-3
166         lmd=0.5*(l2+l1);

```

```

167      xnew=max(0,max(x-mv,min(1,min(x+mv,x.*sqrt(abs(dc)./dv/lmd))))));
168      if ft==1,    xs=xnew;    else,    xs(:)=(H*xnew(:))./Hs;end
169      V=vol();    Vr=V/V0;
170      if Vr>volfrac, l1=lmd; else, l2=lmd; end
171  end
172  ch=max(abs(xnew(:)-x(:))); x=xnew; lt=lt+toc;
173 % PRINT RESULTS
174 pr='It.:%5i Obj.:%11.2f Vol.:%7.3f ch.:%7.3f t.:%7.2f Penal.:%7.2f\n';
175 fprintf(pr,lp,C,Vr,ch,lt,pn); delete(findobj('type', 'patch'));
176 % PLOT DENSITIES
177 o=@(x,y,z) (repmat(linspace(x,y,z)',1,3));
178 O=[o(0,0.1,100);o(0.1,0.3,150);o(0.3,0.5,200);o(0.5,1,650)];
179 P=patch('Faces',ep(:,[1 5 2 6 3 7 4 8 1]),'Vertices',cd,'EdgeColor',...
180 'none','FaceVertexCData',1-xs); caxis([0 1]); daspect([1 1 0.1]);
181 if M, P.FaceColor='flat'; else, P.FaceColor='interp'; end
182 axis off; colormap(O); drawnow;
183 end
%% 3D JACOBIAN FOR PENALIZED THICKNESS
184 function dJp=J3d()
185     tp=(Em+xs(ep).^pn*(E0-Em));
186     Zp(:,1:8)=-(tp)/2; Zp(:,9:12)=0; Zp(:,13:20)=(tp)/2; sZp=ds*Zp';
187     S(7:9:72,:)=sZp(1:3:end,:); S(8:9:72,:)=sZp(2:3:end,:);
188     S(9:9:72,:)=sZp(3:3:end,:); s=reshape(S,3,3,8,et);
189     dJp=s(1,1,:,:).*s(2,2,:,:).*s(3,3,:,:)-s(3,2,:,:).*s(2,3,:,:)-...
190         s(1,2,:,:).*s(2,1,:,:).*s(3,3,:,:)-s(3,1,:,:).*s(2,3,:,:)+...
191         s(1,3,:,:).*s(2,1,:,:).*s(3,2,:,:)-s(3,1,:,:).*s(2,2,:,:));
192     if find(dJp<10*eps, 1), disp('Jacobian<0!'); return; end
193     dJp=dJp(:,:,1:4,:)+dJp(:,:,5:8,:);
194 end
%% VOLUME CHANGES USING ACTUAL THICKNESS
195 function Vp=vol()
196     if ~M
197         tn=Em+xs(ep)*(E0-Em);
198         Z(:,1:8)=-(tn)/2; Z(:,9:12)=0; Z(:,13:20)=(tn)/2; sZ=ds*Z';
199         S(7:9:72,:)=sZ(1:3:end,:); S(8:9:72,:)=sZ(2:3:end,:);
200         S(9:9:72,:)=sZ(3:3:end,:); s=reshape(S,3,3,8,et);
201         dJ=s(1,1,:,:).*s(2,2,:,:).*s(3,3,:,:)-s(3,2,:,:).*s(2,3,:,:)-...
202             s(1,2,:,:).*s(2,1,:,:).*s(3,3,:,:)-s(3,1,:,:).*s(2,3,:,:)+...
203             s(1,3,:,:).*s(2,1,:,:).*s(3,2,:,:)-s(3,1,:,:).*s(2,2,:,:));
204         if find(dJ<10*eps, 1), disp('Jacobian<0!'); return; end
205         Ve=f(sum(dJ,3)); Vp=sum(Ve);
206     else, Vp=sum(xs(:))*ez*ez;
207     end
208 end
209 end
210 end
211 end
212 %
213 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
214 % 2.5D SIMTP code by Tejeswar YARLAGADDA % %
215 % PhD Student, BSE, POLYU, HONGKONG % %
216 % This code is created and modified from top88, initially published by % %
217 % E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov and O. Sigmund, % %
218 % Department of Solid Mechanics, % %
219 % Technical University of Denmark, % %
220 % DK-2800 Lyngby, Denmark. % %
221 % Please send your comments on SIMTP to: yarlagadda.tejeswar@gmail.com % %
222 % Please send your comments on SIMP to: sigmund@fam.dtu.dk % %
223 % % %
224 % Email to yarlagadda.tejeswar@gmail.com for SIMTP code % %
225 % The SIMP code is available at: http://www.topopt.dtu.dk % %

```

```

226 % %
227 % Disclaimer: %
228 % The authors reserve all rights but do not guaranty that the code is %
229 % free from errors. Furthermore, we shall not be liable in any event %
230 % caused by the use of the program. %
231 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Examples:

```

M1: topMBB3D20N([15,4,2.5],[0.19,2.5],-100,[230000,1/3],0.5,3,1.5,0)
M2: topMBB3D20N([15,4,2.5],0.3825,-100,[230000,1/3],0.5,3,1.5,0)
M3: topMBB3D20N([15,4,2.5],[0.1,2.5],-100,[230000,1/3],0.5,3,1.5,0)
M4: topMBB3D20N([15,4,2.5],0.1,-100,[230000,1/3],0.5,3,1.5,1,0)

```

MATLAB topMBB3D20N code:

```

1 function topMBB3D20N(domain,ez,load,matprop,volfrac,penal,rmin,itProc,alp)
2 %% MBB SETUP
3 % Planar direction setup for MBB beam
4 close all; dis=0; f=@(x) reshape(x,[],1);
5 if length(ez)==1, ezX=ez; ezY=ez; ezZ=ez;
6 else, ezX=ez(1); ezY=ez(1); ezZ=ez(2);
7 end
8 a=domain(1); b=domain(2); c=domain(3);
9 nX1=round(a/ezX); nY1=round(b/ezY); nZ1=round(c/ezZ);
10 ezX=a/nX1; ezY=b/nY1; ezZ=c/nZ1; cz=(-nZ1:nZ1)*ezZ/2;
11 [aX,bY]=meshgrid(0:nX1,0:nY1); ccX=aX*ezX; ccY=bY*ezY; nc=(nX1+1)*(nY1+1);
12 ns=reshape(1:nc,nY1+1,[]); cc=zeros(nc,2); cc(ns,:,:)=[ccX(:) ccY(:)];
13 % planar (XY) topology and coordinates
14 c1=f(ns(1:end-1,1:end-1)); c2=f(ns(1:end-1,2:end));
15 c3=f(ns(2:end,2:end)); c4=f(ns(2:end,1:end-1));
16 ec=[c1 c2 c3 c4]; ed=reshape(ec(:,[1:4,2:4,1]),[],2);
17 [en,~,ix]=unique(sort(ed,2), 'rows');
18 em=reshape(ix,[],4)+nc; cm=(cc(en(:,1),:)+cc(en(:,2),:))/2;
19 epP=[ec em]; cdP=[cc;cm]; ntP=nc+size(cm,1); etP=size(epP,1);
20 % Preparing 3D Topology
21 n1=f(repmat((1:ntP),1,1,nZ1+1)+reshape((0:nZ1)*(ntP+nc),1,1,nZ1+1));
22 n2=f(repmat((1:nc)+ntP,1,1,nZ1)+reshape((0:nZ1-1)*(ntP+nc),1,1,nZ1));
23 nt=max([n1;n2]); cd=zeros(nt,3);
24 cd(n1,:)=[repmat(cdP,nZ1+1,1) f(repmat(cZ(1:2:end),ntP,1))];
25 cd(n2,:)=[repmat(cc,nZ1,1) f(repmat(cZ(2:2:end),nc,1))];
26 et=nX1*nY1*nZ1; ep=zeros(et,20);
27 epR=reshape(repmat((0:nZ1-1)*(ntP+nc),etP,1),[],1);
28 ep(:,1:8)=repmat(epP,nZ1,1)+epR; ep(:,9:12)=repmat(ec,nZ1,1)+epR+ntP;
29 ep(:,13:20)=repmat(epP,nZ1,1)+epR+ntP+nc;
30 % Initial Volume
31 V0=a*b*c;
32 %% DEFINE LOADS AND SUPPORTS
33 adfs=(1:3*nt)'; U=zeros(3*nt,1); lel=(1:etP:et)+(0:nY1-1)';
34 ln=unique(f(ep(lel,:),[1,4,8,9,12,13,16,20])); rel=(etP:etP:et)+1-nY1;
35 rn=unique(f(ep(rel,[2,10,14])));
36 fxdfs=[3*ln-2;3*ln;3*rn-1]; fdfs=setdiff(adfs,fxdfs);
37 nlx=4; % Number of elements rows in x-direction for distributing load
38 we=repmat((nlx:-1:1)*load/(sum(1:nlx)*nZ1),nZ1,1);
39 we=we(:).*[-1/12*ones(1,4) 1/3*ones(1,4)]; lel=(0:nZ1-1)'*etP+nY1*(1:nlx);
40 lid=3*f(ep(lel,[4,3,15,16,7,11,19,12]))-1;
41 F=accumarray(lid,we(:),[3*nt,1],[],0,1);
42 %% MATERIAL PROPERTIES AND CONSTITUTIVE MATRIX

```

## Appendix C

```

0.0223,0.0129,0.4072,0.0129,0.0704,0.2629,0.2629,0.0704;0.0223,-0.0739,-0.1796,-
0.0129,0.0516,0.0704,0.1925,-0.0704,-0.0704,0.0704,0.2629,-0.2629,-0.0129,-
0.1796,-0.311,-0.4072,0.1925,0.2629,0.7182,-0.2629;0.1796,0.0739,-0.0223,0.0129,-
0.0704,-0.0516,0.0704,-0.1925,-0.2629,-
0.0704,0.0704,0.2629,0.311,0.1796,0.0129,0.4072,-0.2629,-0.1925,0.2629,-
0.7182;0.1796,0.0739,0.1796,0.311,-0.0704,-0.0704,-0.2629,-0.2629,-0.1925,-
0.0516,-0.1925,-0.7182,0.0129,-0.0223,0.0129,0.4072,0.0704,0.0704,0.2629,0.2629];
60 JTx=dNr*cd(ep(1,:),1); JTy=dNr*cd(ep(1,:),2); JTz=dNr*cd(ep(1,:),3);
61 JT(1:9:72,:) = JTx(1:3:end,:); JT(2:9:72,:) = JTx(2:3:end,:);
62 JT(3:9:72,:) = JTx(3:3:end,:); JT(4:9:72,:) = JTy(1:3:end,:);
63 JT(5:9:72,:) = JTy(2:3:end,:); JT(6:9:72,:) = JTy(3:3:end,:);
64 JT(7:9:72,:) = JTz(1:3:end,:); JT(8:9:72,:) = JTz(2:3:end,:);
65 JT(9:9:72,:) = JTz(3:3:end,:); JT=reshape(JT,3,3,8);
66 dJT=JT(1,1,:,:).*JT(2,2,:,:).*JT(3,3,:,:)-JT(3,2,:,:).*JT(2,3,:,:)-...
67 JT(1,2,:,:).*JT(2,1,:,:).*JT(3,3,:,:)-JT(3,1,:,:).*JT(2,3,:,:)+...
68 JT(1,3,:,:).*JT(2,1,:,:).*JT(3,2,:,:)-JT(3,1,:,:).*JT(2,2,:,:);
69 % Preparing Strain Matrix and Stiffness matrix
70 dN=zeros(3,20,8);
71 for i=1:8, ix=[3*i-2;3*i-1;3*i]; dN(:,:,i)=JT(:,:,i)\dNr(ix,:); end
72 B(1,1:3:60,:)=dN(1,:,:); B(2,2:3:60,:)=dN(2,:,:); B(3,3:3:60,:)=dN(3,:,:);
73 B(4,1:3:60,:)=dN(2,:,:); B(4,2:3:60,:)=dN(1,:,:); B(5,2:3:60,:)=dN(3,:,:);
74 B(5,3:3:60,:)=dN(2,:,:); B(6,3:3:60,:)=dN(1,:,:); B(6,1:3:60,:)=dN(3,:,:);
75 K1=zeros(60,60,8,1); for i=1:8, K1(:,:,i)=B(:,:,i)'*D*B(:,:,i)*dJT(i); end
76 KE = sum(K1,3);
77 % PREPARE FINITE ELEMENT ANALYSIS AND FILTER
78 emt=zeros(et,60); emt(:,3:3:end)=ep*3; emt(:,2:3:end)=ep*3-1;
79 emt(:,1:3:end)=ep*3-2; I=reshape(repmat((1:60),60,1),1,[ ]);
80 J=repmat(1:60,1,60); iK=f(emt(:,I)'); jK=f(emt(:,J)');
81 g=@(x) x(ep); eX=g(cd(:,1)); eY=g(cd(:,2)); eZ=g(cd(:,3));
82 h=@(x) sum(x*1/4.*[-ones(1,4) ones(1,8) -ones(1,4) ones(1,4)],2);
83 eXc=h(eX); eYc=h(eY); eZc=h(eZ); eC=[eXc eYc eZc];
84 rmin = rmin+1e-3; [Id,r]=rangesearch(eC,eC,rmin*max(ezX,ezY));
85 w0=cell2mat(arrayfun(@(i) [i*ones(length(Id{i}),1) Id{i}' (rmin-r{i})' ...
86 ,(1:length(eC))', 'un',0]));
87 H=sparse(w0(:,1),w0(:,2),w0(:,3),max(w0(:,1)),max(w0(:,1)),nzmax(w0(:,3)));
88 Hs=sum(H,2);
89 x=ones(et,1); xs=x; lp=0; lt=0; ch=1; maxloop=50;
90 % Linear system of equations Iterative parameters
91 Vr=sum(x)/et; xld=(a-cd(:,1)); xUy=load*xld.*((3*a^2-xld.^2)/(E*c*b^3));
92 % START ITERATION
93 dch=0.01; dv=ones(et,1)*ezX*ezY*ezZ; dv(:)= H*(dv(:)./Hs); ith=0.5;
94 figure()
95 % START ITERATION
96 while ch > dch && lp<maxloop
97     % PENALTY
98     tic; lp=lp+1; pn=pl+dp*(lp-1); if pn>ph, pn=ph; end
99     % FE-ANALYSIS
100     sK=KE(:)*(Em+xs(:).^pn*(E0-Em)); K=sparse(iK,jK,sK(:)); K=(K+K')/2;
101     % Choose the procedure based on the specifications of the computer
102     if itProc==1
103         if nargin<9, alp=0.1; end
104         % Iterative procedure for relatively large number of nodes
105         K=K(fdःfs,fdःfs); ro=symrcm(K); xU=U; xU(2:3:end)=xUy/Vr;
106         K=K(ro,ro); ro=fdःfs(ro); Fs=F(ro); xUs=xU(ro);
107         % Please change alp value > 0 if iterative procedure fails
108         L=ichol(K,struct('type','ict','droptol',1e-3,'diagcomp',alp));
109         xUs=pcg(K,Fs,1e-4,1000,L,L',xUs); U(ro)=xUs;
110     else
111         % Direct procedure

```

```

112         U(fdः)=K(fdः,fdः)\F(fdः);
113     end
114     %% OBJECTIVE FUNCTION, SENSITIVITY ANALYSIS AND FILTERING/MODIFICATION
115     ce=sum((U(emt)*KE).*U(emt),2); C=sum((Em+xs.^pn*(E0-Em)).*ce);
116     dc=-pn*(E0-Em)*xs.^pn-1.*ce;
117     dc(:)=H*(dc(:)./Hs);
118     %% OPTIMALITY CRITERIA UPDATE OF DESIGN VARIABLES
119     l1=0; l2=1e9; mv=0.2;
120     while (l2-l1)/(l1+l2)>1e-3
121         lmd=0.5*(l2+l1);
122         xnew=max(0,max(x-mv,min(1,min(x+mv,x.*sqrt(abs(dc)./dv/lmd))))));
123         xs(:)=(H*xnew(:))./Hs; V=sum(xs(:))*ezX*ezY*ezZ; Vr=V/V0;
124         if Vr>volfrac, l1=lmd; else, l2=lmd; end
125     end
126     ch=max(abs(xnew(:)-x(:))); x=xnew; lt=lt+toc;
127     %% PRINT RESULTS
128     pr='It.:%5i Obj.:%11.2f Vol.:%7.3f ch.:%7.3f t.:%7.2f Penal.:%7.2f\n';
129     fprintf(pr,l1,C,Vr,ch,lt,pn);
130     %% DISPLAY DENSITIES
131     if dis, mydisp(); end
132 end
133 mydisp();
134 %% PLOT DENSITIES
135 function mydisp()
136     clf; xpl=(xs>ith);
137     epl=ep(xpl,:); xP=reshape(repmat(xs(xpl)',6,1),[],1);
138     P=patch('Faces',reshape(epl,[1 5 2 6 3 7 4 8 1 2 10 14 18 15 11 3 ...
139         6 2 13 17 14 18 15 19 16 20 13 1 9 13 20 16 12 4 8 1 1 5 2 10 ...
140         14 17 13 9 1 4 7 3 11 15 19 16 12 4]),9,[]), 'Vertices',...
141     cd(:,[3,1,2]), 'EdgeColor','none', 'FaceVertexCData',xP);
142     caxis([0 1]); daspect([1 1 1]); P.FaceColor='flat';
143     axis off; colormap(jet); view(60,20); set(gcf,'color','w'); drawnow;
144 end
145 end
146 % =====
147 % === This code was modified from top3d.m by Tejeswar Yarlagadda, ===
148 % === PhD Student, BSE, POLYU, HONGKONG ===
149 % === -----
150 % === The top3d code is intended for educational purposes, and ===
151 % === the details and extensions can be found in the paper: ===
152 % === K. Liu and A. Tovar, "An efficient 3D topology optimization code ===
153 % === written in Matlab", Struct Multidisc Optim,50(6):1175-1196, 2014, ===
154 % === doi:10.1007/s00158-014-1107-x ===
155 % === Please send your suggestions and comments on top3d.m to: ===
156 % === kailiu@iupui.edu ===
157 % === Please send your suggestions and comments on topMBB3D20N.m to: ===
158 % === yarlagadda.tejeswar@gmail.com ===
159 % === -----
160 % === Disclaimer: ===
161 % === The authors reserves all rights for the program. ===
162 % === The code may be distributed and used for educational purposes. ===
163 % === The authors do not guarantee that the code is free from errors ===

```

## References

- [1] Vantyghem G, De Corte W, Shakour E, Amir O. 3D printing of a post-tensioned concrete girder designed by topology optimization. Autom Constr 2020;112. <https://doi.org/10.1016/j.autcon.2020.103084>.
- [2] Zhang Z, Yarlagadda T, Zheng Y, Jiang L, Usmani A. Isogeometric analysis-based design of post-tensioned concrete beam towards construction-oriented topology optimization. Struct Multidisc Optim 2021. <https://doi.org/10.1007/s00158-021-03058-z>
- [3] Bendsøe MP, Kikuchi N. Generating optimal topologies in structural design using a homogenization method. Comput Methods Appl Mech Eng 1988;71:197–224. [https://doi.org/10.1016/0045-7825\(88\)90086-2](https://doi.org/10.1016/0045-7825(88)90086-2)
- [4] Bendsøe MP. Optimal shape design as a material distribution problem. Struct Optim 1989;1:193–202. <https://doi.org/10.1007/BF01650949>
- [5] Zhou M, Rozvany GIN. The COC algorithm, Part II: Topological, geometrical and generalized shape optimization. Comput Methods Appl Mech Eng 1991;89:309–36. [https://doi.org/10.1016/0045-7825\(91\)90046-9](https://doi.org/10.1016/0045-7825(91)90046-9)
- [6] Rozvany GIN, Zhou M, Birker T. Generalized shape optimization without homogenization. Structural Optimization 1992;4:250. <https://doi.org/10.1007/BF01742754>
- [7] Xie YM, Steven GP. A simple evolutionary procedure for structural optimization. Comput Struct 1993;49:885–96. [https://doi.org/10.1016/0045-7949\(93\)90035-C](https://doi.org/10.1016/0045-7949(93)90035-C)
- [8] Sethian JA, Wiegmann A. Structural Boundary Design via Level Set and Immersed Interface Methods. J Comput Phys 2000;163:489–528. <https://doi.org/10.1006/jcph.2000.6581>
- [9] Stolpe M, Svanberg K. An alternative interpolation scheme for minimum compliance topology optimization. Struct Multidisc Optim 2001;22:116–24. <https://doi.org/10.1007/s001580100129>

- [10] Bourdin B, Chambolle A. Design-dependent loads in topology optimization. *ESAIM: COCV* 2003;9:19–48. <https://doi.org/10.1051/cocv:2002070>.
- [11] Bruns TE. A reevaluation of the SIMP method with filtering and an alternative formulation for solid-void topology optimization. *Struct Multidiscip Optim* 2005;30:428–36. <https://doi.org/10.1007/s00158-005-0537-x>.
- [12] Fuchs MB, Jin S, Peleg N. The SRV constraint for 0/1 topological design. *Struct Multidiscip Optim* 2005;30:320–6. <https://doi.org/10.1007/s00158-005-0526-0>.
- [13] Kobayashi MH. On a biologically inspired topology optimization method. *Commun Nonlinear Sci Numer Simul* 2010;15:787. <https://doi.org/10.1016/j.cnsns.2009.04.014>.
- [14] Guo X, Zhang W, Zhong W. Doing Topology Optimization Explicitly and Geometrically—A New Moving Morphable Components Based Framework. *J Appl Mech* 2014;81. <https://doi.org/10.1115/1.4027609>.
- [15] Huang X, Xie YM. Bidirectional Evolutionary Topology Optimization for Structures with Geometrical and Material Nonlinearities. *AIAA Journal* 2007;45:308–13. <https://doi.org/10.2514/1.25046>.
- [16] Díaz A, Sigmund O. Checkerboard patterns in layout optimization. *Struct Optim* 1995;10:40–5. <https://doi.org/10.1007/BF01743693>.
- [17] Jog CS, Haber RB. Stability of finite element models for distributed-parameter optimization and topology design. *Comput Methods Appl Mech Eng* 1996;130:203–26. [https://doi.org/10.1016/0045-7825\(95\)00928-0](https://doi.org/10.1016/0045-7825(95)00928-0).
- [18] Sigmund O. Design of material structures using topology optimization [Doctoral dissertation]. Technical University of Denmark Lyngby 1994. , [https://www.researchgate.net/publication/261173987\\_Design\\_of\\_Material\\_Structures\\_Using\\_Topology\\_Optimization](https://www.researchgate.net/publication/261173987_Design_of_Material_Structures_Using_Topology_Optimization).
- [19] Sigmund O, Petersson J. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural optimization* 1998;16:68–75. <https://doi.org/10.1007/BF01214002>.
- [20] Bourdin B. Filters in topology optimization. *Int J Numer Meth Eng* 2001;50:2143–58. <https://doi.org/10.1002/nme.116>.
- [21] Bruns TE, Tortorelli DA. Topology optimization of non-linear elastic structures and compliant mechanisms. *Comput Methods Appl Mech Eng* 2001;190:3443–59. [https://doi.org/10.1016/S0045-7825\(00\)00278-4](https://doi.org/10.1016/S0045-7825(00)00278-4).
- [22] Poulsen TA. A new scheme for imposing a minimum length scale in topology optimization. *Int J Numer Meth Eng* 2003;57:741–60. <https://doi.org/10.1002/nme.694>.
- [23] Haber RB, Jog CS, Bendsøe MP. A new approach to variable-topology shape design using a constraint on perimeter. *Struct Optim* 1996;11:1–12. <https://doi.org/10.1007/BF01279647>.
- [24] Petersson J, Sigmund O. Slope constrained topology optimization. *Int J Numer Meth Eng* 1998;41:1417–34. [https://doi.org/10.1002/\(SICI\)1097-0207\(19980430\)41:8<1417::AID-NME344>3.0.CO;2-N](https://doi.org/10.1002/(SICI)1097-0207(19980430)41:8<1417::AID-NME344>3.0.CO;2-N).
- [25] Zhou M, Shyy YK, Thomas HL. Checkerboard and minimum member size control in topology optimization. *Struct Multidiscip Optim* 2001;21:152–8. <https://doi.org/10.1007/s00158-001-0517-9>.
- [26] Guest JK, Prévost JH, Belytschko T. Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *Int J Numer Meth Eng* 2004;61:238–54. <https://doi.org/10.1002/nme.1064>.
- [27] Sigmund O. Morphology-based black and white filters for topology optimization. *Struct Multidiscip Optim* 2007;33:401–24. <https://doi.org/10.1007/s00158-006-0087-x>.
- [28] Jang G-W, Jeong JH, Kim YY, Sheen D, Park C, Kim M-N. Checkerboard-free topology optimization using non-conforming finite elements. *Int J Numer Meth Eng* 2003;57:1717–35. <https://doi.org/10.1002/nme.738>.
- [29] Jang G-W, Lee S, Kim YY, Sheen D. Topology optimization using non-conforming finite elements: three-dimensional case. *Int J Numer Meth Eng* 2005;63:859–75. <https://doi.org/10.1002/nme.1302>.
- [30] Liu Z, Korvink JG. Adaptive moving mesh level set method for structure topology optimization. *Eng Optim* 2008;40:529–58. <https://doi.org/10.1080/03052150801985544>.
- [31] Wang Y-Q, He J-J, Luo Z, Kang Z. An adaptive method for high-resolution topology design. *Acta Mech Sin* 2013;29:840–50. <https://doi.org/10.1007/s10409-013-0084-4>.
- [32] Wang Y, Kang Z, He Q. An adaptive refinement approach for topology optimization based on separated density field description. *Comput Struct* 2013;117:10–22. <https://doi.org/10.1016/j.compstruc.2012.11.004>.
- [33] Wang H, Liu J, Wen G. An adaptive mesh-adjustment strategy for continuum topology optimization to achieve manufacturable structural layout. *Int J Numer Meth Eng* 2019;117:1304–22. <https://doi.org/10.1002/nme.6001>.
- [34] Kumar AV, Gossard DC. Synthesis of Optimal Shape and Topology of Structures. *J Mech Des* 1996;118:68–74. <https://doi.org/10.1115/1.2826858>.
- [35] Bendsøe MP, Sigmund O. Topology optimization by distribution of isotropic material. *Topology Optimization: Theory, Methods, and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2004. p. 1–69. [https://doi.org/10.1007/978-3-662-05086-6\\_1](https://doi.org/10.1007/978-3-662-05086-6_1).
- [36] Hammer V. Checkmate? Nodal densities in topology optimization. Proc II Max Planck Workshop on Engineering Design Optimization, Dept of Mathematics, DTU, Denmark2001. Retrieved.
- [37] Belytschko T, Xiao SP, Parimi C. Topology optimization with implicit functions and regularization. *Int J Numer Meth Eng* 2003;57:1177–96. <https://doi.org/10.1002/nme.824>.
- [38] Matsui K, Terada K. Continuous approximation of material distribution for topology optimization. *Int J Numer Meth Eng* 2004;59:1925–44. <https://doi.org/10.1002/nme.945>.
- [39] Guest JK. Topology optimization with multiple phase projection. *Comput Methods Appl Mech Eng* 2009;199:123–35. <https://doi.org/10.1016/j.cma.2009.09.023>.
- [40] Guest JK, Genut LCS. Reducing dimensionality in topology optimization using adaptive design variable fields. *Int J Numer Meth Eng* 2010;81:1019–45. <https://doi.org/10.1002/nme.2724>.
- [41] Rahmatalla SF, Swan CC. A Q4/Q4 continuum structural topology optimization implementation. *Struct Multidiscip Optim* 2004;27:130–5. <https://doi.org/10.1007/s00158-003-0365-9>.
- [42] Paulino GH, Le CH. A modified Q4/Q4 element for topology optimization. *Struct Multidiscip Optim* 2009;37:255–64. <https://doi.org/10.1007/s00158-008-0228-5>.
- [43] Kang Z, Wang Y. Structural topology optimization based on non-local Shepard interpolation of density field. *Comput Methods Appl Mech Eng* 2011;200:3515–25. <https://doi.org/10.1016/j.cma.2011.09.001>.
- [44] Kang Z, Wang Y. A nodal variable method of structural topology optimization based on Shepard interpolant. *Int J Numer Meth Eng* 2012;90:329–42. <https://doi.org/10.1002/nme.3321>.
- [45] Nguyen TH, Paulino GH, Song J, Le CH. A computational paradigm for multiresolution topology optimization (MTOPO). *Struct Multidiscip Optim* 2010;41:525–39. <https://doi.org/10.1007/s00158-009-0443-8>.
- [46] Almeida SRM, Paulino GH, Silva ECN. Layout and material gradation in topology optimization of functionally graded structures: a global-local approach. *Struct Multidiscip Optim* 2010;42:855–68. <https://doi.org/10.1007/s00158-010-0514-x>.
- [47] Areias P, Rodrigues HC, Rabczuk T. Coupled finite-element/topology optimization of continua using the Newton-Raphson method. *Eur J Mech A Solids* 2021;85. <https://doi.org/10.1016/j.euromechsol.2020.104117>.
- [48] Lambe AB, Czekanski A. Adaptive Topology Optimization Using a Continuous Approximation of Material Distribution. ASME 2016 International Mechanical Engineering Congress and Exposition. Arizona, USA2016. <https://doi.org/10.1115/imece2016-65537>. Retrieved 5/23/2021.
- [49] Lambe AB, Czekanski A. Topology optimization using a continuous density field and adaptive mesh refinement. *Int J Numer Meth Eng* 2018;113:357–73. <https://doi.org/10.1002/nme.5617>.
- [50] Gao J, Xue H, Gao L, Luo Z. Topology optimization for auxetic metamaterials based on isogeometric analysis. *Comput Methods Appl Mech Eng* 2019;352:211–36. <https://doi.org/10.1016/j.cma.2019.04.021>.
- [51] Gao J, Gao L, Luo Z, Li P. Isogeometric topology optimization for continuum structures using density distribution function. *Int J Numer Meth Eng* 2019;119:991–1017. <https://doi.org/10.1002/nme.6081>.
- [52] Gao J, Luo Z, Xiao M, Gao L, Li P. A NURBS-based Multi-Material Interpolation (N-MMI) for isogeometric topology optimization of structures. *Appl Math Model* 2020;81:818–43. <https://doi.org/10.1016/j.apm.2020.01.006>.
- [53] Cho S, Kwak J. Topology design optimization of geometrically non-linear structures using meshfree method. *Comput Methods Appl Mech Eng* 2006;195:5909–25. <https://doi.org/10.1016/j.cma.2005.08.015>.
- [54] Zhou JX, Zou W. Meshless approximation combined with implicit topology description for optimization of continua. *Struct Multidiscip Optim* 2008;36:347–53. <https://doi.org/10.1007/s00158-007-0168-5>.
- [55] Du Y, Luo Z, Tian Q, Chen L. Topology optimization for thermo-mechanical compliant actuators using mesh-free methods. *Eng Optim* 2009;41:753–72. <https://doi.org/10.1080/03052150902834989>.
- [56] Long K, Zuo Z, Zuberi RH. Study on parameters for topological variables field interpolated by moving least square approximation. *Acta Mech Solidi Sin* 2009;22:180–8. [https://doi.org/10.1016/S0894-9166\(09\)60103-8](https://doi.org/10.1016/S0894-9166(09)60103-8).
- [57] Luo Z, Zhang N, Wang Y, Gao W. Topology optimization of structures using meshless density variable approximants. *Int J Numer Meth Eng* 2013;93:443–64. <https://doi.org/10.1002/nme.4394>.
- [58] Wang Y, Luo Z, Wu J, Zhang N. Topology optimization of compliant mechanisms using element-free Galerkin method. *Adv Eng Softw* 2015;85:61–72. <https://doi.org/10.1016/j.advengsoft.2015.03.001>.
- [59] Zhang Y, Ge W, Tong X, Ye M. Topology optimization of structures with coupled finite element – Element-free Galerkin method. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 2018;232:731–45. <https://doi.org/10.1177/0954406216688716>.
- [60] Chi H, Pereira A, Menezes IFM, Paulino GH. Virtual element method (VEM)-based topology optimization: an integrated framework. *Struct Multidiscip Optim* 2020;62:1089–114. <https://doi.org/10.1007/s00158-019-02268-w>.
- [61] Yang D, Liu H, Zhang W, Li S. Stress-constrained topology optimization based on maximum stress measures. *Comput Struct* 2018;198:23–39. <https://doi.org/10.1016/j.compstruc.2018.01.008>.
- [62] Zhu B, Zhang X, Li H, Liang J, Wang R, Li H, et al. An 89-line code for geometrically nonlinear topology optimization written in FreeFEM. *Struct Multidiscip Optim* 2021;63:1015–27. <https://doi.org/10.1007/s00158-020-02733-x>.
- [63] Zhao F. A nodal variable ESO (BESO) method for structural topology optimization. *Finite Elem Anal Des* 2014;86:34–40. <https://doi.org/10.1016/j.finel.2014.03.012>.
- [64] Kogiso N, Ahn W, Nishiwaki S, Izui K, Yoshimura M. Robust Topology Optimization for Compliant Mechanisms Considering Uncertainty of Applied Loads. *Journal of Advanced Mechanical Design, Systems, and Manufacturing* 2008;2:96. <https://doi.org/10.1299/jamsdm.2.96>.
- [65] Kobayashi M, Nishiwaki S, Izui K, Yoshimura M. An innovative design method for compliant mechanisms combining structural optimisations and designer

- creativity. *J Eng Des* 2009;20:125–54. <https://doi.org/10.1080/09544820701565017>.
- [66] Zhan J, Yang K, Huang K. Topological Design of Hinge-Free Compliant Mechanisms Using the Node Design Variables Method. In: Zhang X, Liu H, Chen Z, Wang N, editors. Intelligent Robotics and Applications. Cham: Springer International Publishing; 2014. p. 567–75. [https://doi.org/10.1007/978-3-319-13963-0\\_57](https://doi.org/10.1007/978-3-319-13963-0_57). Retrieved.
- [67] Zhan J, Long L, Huang Z. Topology Optimization of Thermally Actuated Compliant Mechanisms Using Node Design Variables. In: Zhang X, Wang N, Huang Y, editors. Mechanism and Machine Science. Singapore: Springer Singapore; 2016. p. 667–75. [https://doi.org/10.1007/978-981-10-2875-5\\_55](https://doi.org/10.1007/978-981-10-2875-5_55). Retrieved.
- [68] Du J, Olhoff N. Topological optimization of continuum structures with design-dependent surface loading – Part I: new computational approach for 2D problems. *Struct Multidiscip Optim* 2004;27:151–65. <https://doi.org/10.1007/s00158-004-0379-v>.
- [69] Du J, Olhoff N. Topological optimization of continuum structures with design-dependent surface loading – Part II: algorithm and examples for 3D problems. *Struct Multidiscip Optim* 2004;27:166–77. <https://doi.org/10.1007/s00158-004-0380-5>.
- [70] Carbonari RC, Silva ECN, Nishiwaki S. Topology optimization applied to the design of multiactuated piezoelectric microtools. *Smart Structures and Materials 2004: Modeling, Signal Processing, and Control: International Society for Optics and Photonics*; 2004. p. 277–88. <https://doi.org/10.1117/12.538584>. Retrieved.
- [71] Carbonari RC, Silva EC, Nishiwaki S. Optimum placement of piezoelectric material in piezoactuator design. *Smart Mater Struct* 2007;16:207. <https://doi.org/10.1088/0964-1726/16/1/025>.
- [72] Wang H, Liu J, Wen G. An efficient multi-resolution topology optimization scheme for stiffness maximization and stress minimization. *Eng Optim* 2020;1–21. <https://doi.org/10.1080/0305215X.2020.1853713>.
- [73] Rozvany GIN. A critical review of established methods of structural topology optimization. *Struct Multidiscip Optim* 2009;37:217–37. <https://doi.org/10.1007/s00158-007-0217-0>.
- [74] Sigmund O, Maute K. Topology optimization approaches. *Struct Multidiscip Optim* 2013;48:1031–55. <https://doi.org/10.1007/s00158-013-0978-6>.
- [75] Deaton JD, Grandhi RV. A survey of structural and multidisciplinary continuum topology optimization: part 2000. *Struct Multidiscip Optim* 2014;49:1–38. <https://doi.org/10.1007/s00158-013-0956-z>.
- [76] Zegard T, Hartz C, Mazurek A, Baker WF. Advancing building engineering through structural and topology optimization. *Struct Multidiscip Optim* 2020;62:915–35. <https://doi.org/10.1007/s00158-020-02506-6>.
- [77] Van Dijk NP, Maute K, Langelaar M, van Keulen F. Level-set methods for structural topology optimization: a review. *Struct Multidiscip Optim* 2013;48:437–72. <https://doi.org/10.1007/s00158-013-0912-y>.
- [78] Zhu J-H, Zhang W-H, Xia L. Topology Optimization in Aircraft and Aerospace Structures Design. *Arch Comput Methods Eng* 2016;23:595–622. <https://doi.org/10.1007/s11831-015-9151-2>.
- [79] Zhu B, Zhang X, Zhang H, Liang J, Zang H, Li H, et al. Design of compliant mechanisms using continuum topology optimization: A review. *Mech Mach Theory* 2020;143:. <https://doi.org/10.1016/j.mechmachtheory.2019.103622>103622.
- [80] Plocher J, Panesar A. Review on design and structural optimisation in additive manufacturing: Towards next-generation lightweight structures. *Mater Des* 2019;183;. <https://doi.org/10.1016/j.matdes.2019.108164>108164.
- [81] Rossow MP, Taylor JE. A Finite Element Method for the Optimal Design of Variable Thickness Sheets. *AIAA Journal* 1973;11:1566–9. <https://doi.org/10.2514/3.50631>.
- [82] Bendsøe MP. Optimization of structural topology, shape, and material. 1 ed: Springer; 1995. <https://doi.org/10.1007/978-3-662-03115-5>.
- [83] Li Q, Steven GP, Xie Y. Evolutionary thickness design with stiffness maximization and stress minimization criteria. *Int J Numer Meth Eng* 2001;52:979–95. <https://doi.org/10.1002/nme.241>.
- [84] Li Q, Steven GP, Xie Y. Evolutionary structural optimization for stress minimization problems by discrete thickness design. *Comput Struct* 2000;78:769–80. [https://doi.org/10.1016/S0045-7949\(00\)00057-2](https://doi.org/10.1016/S0045-7949(00)00057-2).
- [85] Li Q, Steven GP, Xie Y. A simple checkerboard suppression algorithm for evolutionary structural optimization. *Struct Multidiscip Optim* 2001;22:230–9. <https://doi.org/10.1007/s001580100140>.
- [86] Makrodimopoulos A, Bhaskar A, Keane AJ. A formulation of thickness optimization for plane stress. Proceedings of 17th UK Conference on Computational Mechanics (ACME-UK) (05/04/09 - 07/04/09)2009. <https://eprints.soton.ac.uk/69907/>.
- [87] Makrodimopoulos A, Bhaskar A, Keane AJ. Second-order cone programming formulations for a class of problems in structural optimization. *Struct Multidiscip Optim* 2010;40:365. <https://doi.org/10.1007/s00158-009-0376-2>.
- [88] Kennedy CJ. Discrete thickness optimization via piecewise constraint penalization. *Struct Multidiscip Optim* 2015;51:1247–65. <https://doi.org/10.1007/s00158-014-1210-z>.
- [89] Dvorkin EN, Bathe KJ. A continuum mechanics based four-node shell element for general non-linear analysis. *Engineering Computations* 1984;1:77–88. <https://doi.org/10.1108/eb023562>.
- [90] Shabana AA, Yakoub RY. Three Dimensional Absolute Nodal Coordinate Formulation for Beam Elements: Theory. *J Mech Des* 2000;123:606–13. <https://doi.org/10.1115/1.1410100>.
- [91] Mikkola AM, Shabana AA. A Non-Incremental Finite Element Procedure for the Analysis of Large Deformation of Plates and Shells in Mechanical System Applications. *Multibody SysDyn* 2003;9:283–309. <https://doi.org/10.1023/A:1022950912782>.
- [92] Abbas LK, Rui X, Hammoudi ZS. Plate/shell element of variable thickness based on the absolute nodal coordinate formulation. Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics 2010;224:127–41. <https://doi.org/10.1243/14644193imbd244>.
- [93] Ko Y, Lee P-S, Bathe K-J. The MITC4+ shell element and its performance. *Comput Struct* 2016;169:57–68. <https://doi.org/10.1016/j.compstruc.2016.03.002>.
- [94] Zienkiewicz OC, Taylor RL, Zhu JZ. Chapter 8 - The Patch Test, Reduced Integration, and Nonconforming Elements. In: Zienkiewicz OC, Taylor RL, Zhu JZ, editors. *The Finite Element Method: its Basis and Fundamentals* (Seventh Edition). Oxford: Butterworth-Heinemann; 2013. p. 257–84. <https://doi.org/10.1016/B978-1-85617-633-0-00008-3>.
- [95] Shepard D. A two-dimensional interpolation function for irregularly-spaced data. In: Proceedings of Proceedings of the 1968 23rd ACM national conference: Association for Computing Machinery. <https://doi.org/10.1145/800186.810616>.
- [96] Bucki M, Lobos C, Payan Y, Hitschfeld N. Jacobian-based repair method for finite element meshes after registration. *Engineering with Computers* 2011;27:285–97. <https://doi.org/10.1007/s00366-010-0198-2>.
- [97] Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O. Efficient topology optimization in MATLAB using 88 lines of code. *Struct Multidiscip Optim* 2011;43:1–16. <https://doi.org/10.1007/s00158-010-0594-7>.
- [98] Liu K, Tovar A. An efficient 3D topology optimization code written in Matlab. *Struct Multidiscip Optim* 2014;50:1175–96. <https://doi.org/10.1007/s00158-014-1107-x>.
- [99] Sven. stlwrite - write ASCII or Binary STL files. MATLAB Central File Exchange: MathWorks; 2021. <https://www.mathworks.com/matlabcentral/fileexchange/20922-stlwrite-write-ascii-or-binary-stl-files>. Retrieved December 8; 2021.