The Visual Computer
https://doi.org/10.1007/s00371-020-01860-2

ORIGINAL ARTICLE

A novel application framework for self-supporting topology optimization

Dengyang Zhao[1] · Ming Li[1] · Yusheng Liu[1]

Abstract

This paper presents an application framework that provides a complete process to design an optimized *self-supporting structure*, ready to be fabricated via additive manufacturing without the usage of additional *support* structures. Such supports in general have to be created during the fabricating process so that the primary object can be manufactured layer by layer without collapse; this process is very time-consuming and waste of material. The main approach resolves this issue by formulating the self-supporting requirements as an explicit quadratic continuous constraint in a topology optimization problem, or specifically, requiring the number of unsupported elements (in terms of the sum of squares of their densities) to be zero. Under the formulation, the required sensitivity of the self-supporting constraint with respect to the design density can be derived straightforward and is only linearly dependent on the density of the element itself. In addition, a novel discrete convolution operator is particularly designed to detect the unsupported elements. The approach works for cases of general overhang angles, and the produced optimized structures have close target compliance to those of the reference structures obtained without considering the self-supporting constraint, as demonstrated by various 2D and 3D benchmark examples.

Keywords Self-supporting · Topology optimization · Explicit quadratic constraints · Additive manufacturing · Discrete convolution

1 Introduction

Topology optimization aims to generate an optimal material distribution within a design domain under certain geometric or physical constraints. Since its introduction in late 1980s [1], this problem has attracted wide industrial and academic interest due to its large potentiality in engineering applications and its intrinsic mathematical challenges. Topology optimization has developed in many different forms, such as: homogenization [1], density (SIMP) [2], evolutionary approaches (BESO) [3,4], level set [5,6], or more recently IGA (iso-geometric analysis) [7,8], to name a few. See also [9] for a recent and comprehensive review on this topic.

The complex geometric designs produced by topology optimization show the approach's superiority in balancing the geometric distribution and the target physical performance. Such designs are, however, very difficult to be manufactured directly via traditional subtractive or formative manufacturing techniques [10–12]. On the other hand, the rapidly developing additive manufacturing technologies have the promise to overcome the barrier between the potentiality that the topology optimization approaches can provide and the limitations that the traditional manufacturing technologies can fabricate. In reality, additive manufacturing is a natural counterpart to topology optimization in that they have very versatile capability to quickly generate and realize new components not existing before [13,14].

Despite the enhanced geometric freedom associated with additive manufacturing, specific design rules must still be satisfied in order to ensure manufacturability. The fabrication overhang angle is such a rule of paramount importance so that the part will not collapse when being fabricating layer by layer. A structure satisfying such an overhang angle constraint is called *self-supporting*. For example, Thomas [15] identified 45° as the typical maximum overhang angle with a large number of experiments. For a non self-supporting structure, its geometry has to be modified or additional *support* structures need to be generated. Modifying the geometry

✉ Ming Li
liming@cad.zju.edu.cn

1 State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, China

Published online: 23 June 2020

will ultimately reduce the structure's physical performance. Additional support raises the issue of automatic and minimum volume support design [16–18], and post-processing to remove the unwanted supports. In the case that the support is made of the same material as the main component, such as the selective laser melting (SLM) process using metals, it is difficult and time-consuming to remove the support structure. Particularly, when the generated supports are embedded within a closed volume of the model, it is impossible to remove them.

The best strategy to resolve the issue is perhaps to design a completely self-supporting structure that can be fabricated directly without the usage of support materials. Brackett et al. first suggested including the overhang angle constraints into the topology optimization process [19], but no self-supporting structure was produced. The first 2D self-supporting structure built from topology optimization is due to the work of Gaynor and Guest in 2014 [20] (and a very recent journal version [21]), achieved via introducing a wedge-shaped filter during the optimization process. Later on, Langelaar [13,14] conducted an intensive study on the topic both in 2D and 3D cases, where 3D self-supporting structures were also generated [14]. Recently, an impressive work was conducted by Qian [22], where the self-supporting constraint is formulated using a novel density gradient-based integral approach. Various impressive 2D and 3D self-supporting structures were produced. Effect of the approach [22] is not explicitly dependent on the elemental grayness but suffers from boundary oscillations and requires relatively large filters. The topic recently has attracted wide research interest, and various other types of constraints were proposed, for example via an edge detection algorithm [23] or as a layer-by-layer mechanical constraint in a purely geometric manner [24]. A concept of 'support structure topological sensitivity' was also introduced by Mirzendehdel [25] to reduce support structures. Furthermore, designing support-free interior material layout was also studied by various researchers. For example, Lee et al [26] proposed a technique of support-free elliptic hollowing, Wang et al. [27] proposed a layer-based hollowing operator, and Xie and Chen recently generated support-free interior carving [28]. In addition, Wu et al. also proposed a novel type of rhombic infill structures and optimized them to generate self-supporting structures [29]. Performance of the approach was recently improved based on a multiresolution optimization approach [30]. We also notice that a novel support-free fabrication approach by multiaxis motion was designed by Dai et al. [31].

In this paper, a novel application framework of self-supporting topology optimization is proposed to generate a structure of optimized physical performance. Based on our definition of the unsupported elements, the self-supporting constraint here is formulated as an *explicit* quadratic function with respect to the design density. It requires the number of unsupported elements (in terms of the sum of their densities) to be zero. The main purpose of the framework is to generate a 0–1 self-supporting structure with close compliance with the original structure. In order to gain a close compliance, several strategies are proposed, in particular, choosing print directions and volume reduction. The sensitivity of self-supporting constraint for each element can be derived straightforwardly and is only linearly dependent on density of the element itself due to the novel quadratic formulation. In addition, a novel and efficient discrete convolution operator in detecting the unsupported elements is created. The framework works for cases of general overhang angles, and the final optimized structures have close target compliances with the referred one without obtaining the self-supporting constraint, as demonstrated by various 2D and 3D benchmark examples.

The remainder of the paper is organized as follows. The novel formulation of self-supporting topology optimization and the framework overview are presented in Sect. 2. Numerical techniques behind the approach are detailed in Sect. 3. Extensive 2D and 3D examples are demonstrated in Sect. 4. Finally, the paper is concluded in Sect. 5.

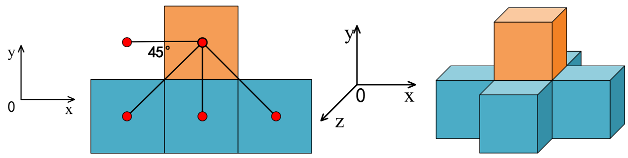
2 Problem statement and framework overview

In the section, the self-supporting constraint is formulated as a quadratic continuous function in terms of the element density in a classical SIMP method [2]. Considering the self-supporting constraint, the compliance of the optimized structure has to be as close as possible with the original structure(without self-supporting constraint). Following on from this, the proposed framework to resolve the problem is outlined, and some particular considerations behind the issue are particularly detailed.

2.1 Supported and unsupported elements

The *supported elements* generally stand for the structural elements that can be fabricated via an additive manufacturing technology without collapse with respect to the fabrication process. They are defined here using the concepts of a maximum printable supporting angle, or overhang angle, which is first assumed to be 45° following the previous study [19]. Extensions of the approach to general overhang angles are also explained later. We also assume that the print direction is following the positive y-axis direction in both 2D and 3D for ease of explanation.

First, consider a 2D discrete structured mesh model $\mathcal{M}$ consisting of square elements $e(n, m)$, that is,



(a) 2D case **(b)** 3D case

Fig. 1 A supported element (in orange) is supported by one of the supporting elements in blue in 2D and 3D

$$\mathcal{M} = \{e(n, m)|\ 1 \leq n \leq N,\ 1 \leq m \leq M\}, \tag{1}$$

where n, m are the indices increasing along the x- and y-axes, respectively. Without confusion, we also use e to represent a square element without explicitly mentioning its indices n, m. In addition, a density matrix $\boldsymbol{\rho}$ of size $s = N \times M$ is also associated with $\mathcal{M}$, where an entry value $\rho(n, m) = 1$ or 0, respectively, represents a solid or void element $e(n, m)$ of $\mathcal{M}$.

As illustrated in Fig. 1a, given a solid element $e(n, m)$ in $\mathcal{M}$ (in orange), it is *supported*, or called a *supported element*, if one of the three blue elements below it is solid. We formulate the self-supporting condition in a continuous form as follows: an element $e(n, m) \in \mathcal{M}$ is supported if

$$\sum_{n-1 \leq r \leq n+1} \rho(r, m-1) > 0. \tag{2}$$

Correspondingly, the *supporting set* $\mathcal{M}_S$ of model $\mathcal{M}$ is the set of all supported elements within $\mathcal{M}$, that is,

$$\mathcal{M}_S = \{e(n, m) \in \mathcal{M}| m = 1 \text{ or } \sum_{n-1 \leq r \leq n+1} \rho(r, m-1) > 0\}. \tag{3}$$

Similarly, given a 3D structured mesh model $\mathcal{M}$ consisting of cubic elements,

$$\mathcal{M} = \{e(n, m, l)\ |\ 1 \leq n \leq N,\ 1 \leq m \leq M,\ 1 \leq l \leq L\}, \tag{4}$$

where n, m, l are the indices increasing along the $x-$, $y-$, z-axes respectively, the supporting set of $\mathcal{M}$ is similarly defined (see also Fig. 1b):

$$\mathcal{M}_S = \{e(n, m, l) \in \mathcal{M}\ |\ m = 1 \text{ or } \sum_{\substack{n-1 \leq r \leq n+1, s=l \\ \text{or } r=n, l-1 \leq s \leq l+1}} \rho(r, m-1, s) > 0\}. \tag{5}$$

Note also that only five elements are included here as other elements do not form appropriate overhang angles with the orange element.

Correspondingly, the set of unsupported elements of a model $\mathcal{M}$ is

$$\mathcal{M}_U = \mathcal{M} \backslash \mathcal{M}_S. \tag{6}$$

2.2 Formulation of self-supporting topology optimization

The self-supporting topology optimization problem aims to find an optimized material distribution within a design domain under certain boundary conditions. As widely studied before, the problem of minimum compliance or equivalently maximum stiffness is examined here. Following the classical SIMP framework [2], the problem of self-supporting topology optimization is formulated here as an optimization problem with an additional explicit self-supporting constraint. The constraint is reformulated using a simple quadratic function with respect to the density, specifically, requiring the number of unsupported elements (in terms of the sum of square of their densities) to be zero. Details are explained below.

The problem of self-supporting topology optimization is stated as: find density distribution $\boldsymbol{\rho}$,

$$\min_{\boldsymbol{\rho} \in \mathbb{R}^{N \times M}} c(\mathbf{u}, \boldsymbol{\rho}),\ s.t. \tag{7}$$

$$\begin{cases} \mathbf{K}(\boldsymbol{\rho})\mathbf{u} = \mathbf{f}(\boldsymbol{\rho}), \\ V(\boldsymbol{\rho})/V_0 \leq f, \\ U(\boldsymbol{\rho}) = \sum_{e \in \mathcal{M}_U} \rho_e^2 \leq \epsilon, \\ 0 < \rho_e \leq 1,\ e = 1, \ldots, s, \end{cases}$$

where $\boldsymbol{\rho}$ is the vector of design variables (element densities) to be computed, $\mathbf{u}$ is the vector of global displacements and $\mathbf{K}$ is the global stiffness matrix. The objective function $c(\mathbf{u}, \boldsymbol{\rho})$ is the structure's compliance, defined as

$$c(\mathbf{u}, \boldsymbol{\rho}) = \mathbf{u}^T \mathbf{K}\mathbf{u}. \tag{8}$$

$\mathbf{f}(\boldsymbol{\rho})$ is the nodal force vector, $V(\boldsymbol{\rho})$ and V_0 are the material volume and design domain volume, f is the prescribed volume fraction, $\mathcal{M}_U$ is the index set of unsupported elements as defined in (6) and $\epsilon > 0$ is a small parameter close to 0. A penalty parameter p, usually set as $p = 3$, is applied here for the 0,1 convergence of $\boldsymbol{\rho}$, or specifically,

$$\mathbf{K}_e = \rho_e^p \mathbf{K}_e^0, \tag{9}$$

where $\mathbf{K}_e^0$ is the element stiffness matrix associated with an element e in the model $\mathcal{M}$ and ρ_e the associated element density.

The only difference between the above conventions in (7) with previous SIMP-based formulations is that it has an additional constraint $U(\boldsymbol{\rho}) = \sum_{e \in \mathcal{M}_U} \rho_e^2 \leq \epsilon$ to meet the self-supporting requirement.

Note here that the self-supporting constraint stated in (7) is based on the observation that when the sum of the element densities of unsupported elements tends to 0, all the elements are self-supported. The simple quadratic expression allows for a straightforward sensitivity derivation of the self-supporting constraint and ultimately results in a linear sensitivity expression.
The self-supporting topology optimization problem (7) is solved using the MMA approach noticing that the sensitivity can be derived straightforward. The key techniques behind it are to be further explained in Sect. 3.1.

In this section, we first describe the overall framework and skills in controlling the overall optimization process. In order for a reliable structure to be generated, the overall optimization process is carefully designed, as shown in Algorithm I. It mainly consists of three main suboptimization processes (C_1, C_2, C_3) for a reliable structure to be generated:

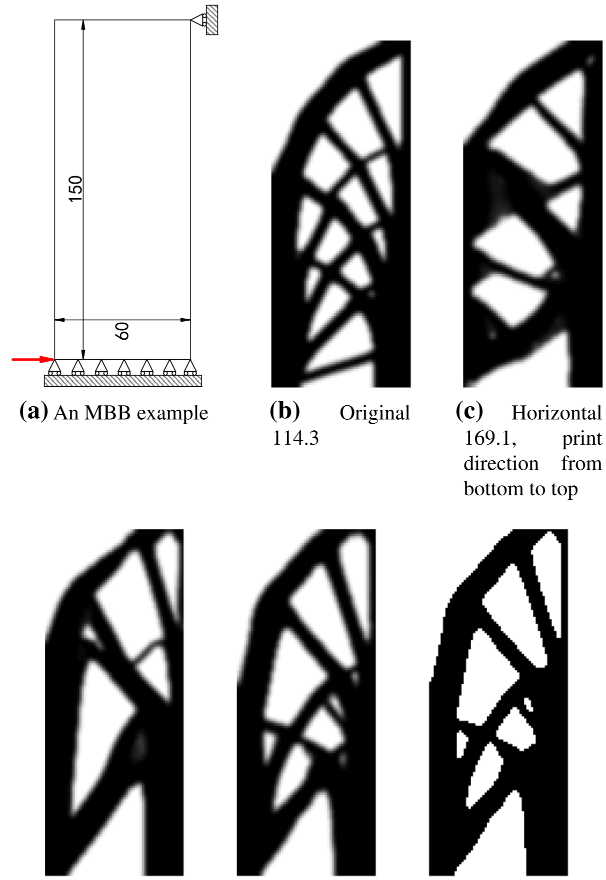
C_1, case of gray elements without self-supporting constraints;

C_2, case of gray elements with self-supporting constraints;

C_3, case of pure 0–1 elements with self-supporting constraints.

Further illustration of C_1, C_2, C_3 is also referred to Fig. 13.

The overall framework is described in **Algorithm 1**, and the above-described three cases are embedded within a recursive volume reduction process so that the self-supporting constraints can be resolved from a larger to smaller design spaces. The main purpose of the framework is to generate a 0–1 self-supporting structure with close compliance with the original structure. In order to gain close compliance, print direction is chosen and volume reduction strategy is used during the optimization. Not just these points, particular considerations on some aspects of the framework are further explained below.



(a) An MBB example **(b)** Original 114.3 **(c)** Horizontal 169.1, print direction from bottom to top **(d)** Vertical 123.2, print direction from right to left **(e)** Volume reduction 119.0 **(f)** Volume reduction with 0-1 tranction 117.2

Fig. 2 Various generated structures for a classical MBB example in **a** (size 150 × 60, filter radius 2.5 and target volume fraction 0.6). **b** was obtained without self-supporting constraints; **c**–**f** with self-supporting constraint; **e**, **f** further uses an iterative volume reduction strategy

2.2.1 C_1 to C_2: choose print direction

Whether an element is self-supporting is dependent on the print direction. In the study of Qian [22] and Langelaar [32], we know that different print directions produce different optimized structures. We here take the optimized structure without self-supporting constraint as benchmark and aim to produce a structure of close shape and close compliance to it. See, for example, in Fig. 2a–c; different structures of different compliances were produced in the case of different print directions. This also shows that it is very important to choose the print direction.

Thus, the direction is determined in the process of computing the optimized structure without self-supporting constraints. The process stops when the grayness of the computed structure reaches certain value, measured using the following defined M_{nd}^0,

$$M_{nd} < M_{nd}^0, \tag{10}$$

where

$$M_{nd} = \frac{\sum_{e=1}^{s} 4\rho_e(1-\rho_e)}{s} \times 100\%, \tag{11}$$

and s is the number of elements of the domain, ρ_e is the density of an element e.

We also plot in Fig. 3 different grayscale structures corresponding to different values of M_{nd} for the MBB example 0.5, 0.4, 0.3, 0.2, respectively. Different occupations of grayscale regions can be observed from the examples. In practice, we set the threshold of M_{nd} as 0.36, which corresponds to a structure of uniform density of 0.1 or 0.9. When

(a) $M_{nd} = 0.5$ **(b)** $M_{nd} = 0.4$ **(c)** $M_{nd} = 0.3$ **(d)** $M_{nd} = 0.2$

Fig. 3 Different values of M_{nd} involved in Eq. (10) indicate different grayness distributions of structures

M_{nd} reaches 0.36,, we detect the unsupported elements in each direction and choose the print direction with least unsupported elements.

2.2.2 C_2 to C_3: from gray density to 0–1 density

During the optimization in the case of C_2, a gray self-supporting structure is produced, which, however, may still have some unsupported elements of very small density values. The gray structure is a relaxation that the definition of unsupported elements is based on 0–1 element. To ensure the accuracy of the self-supporting constraint, the final structure should be directly 0–1. In order to produce a pure 0–1 self-supporting structure, the case C_3 is further studied. In each step of it, a simple density truncation method is applied to set the structure pure 0–1 values satisfying the volume constraint. These values are taken as inputs to the next MMA optimizer.

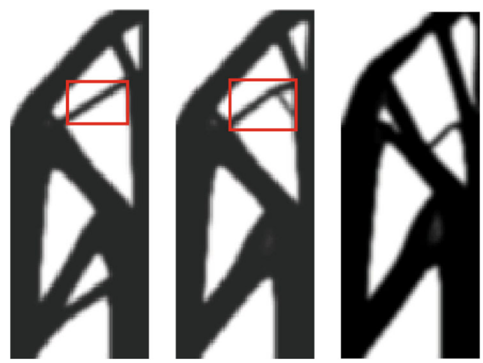
2.2.3 The value of ϵ of self-supporting constraint

The conversion process is controlled using the value of ϵ in **Algorithm 1**, which describes self-supporting situation of a structure. For example, in Fig. 4, different values of ϵ, respectively, 0.1, 0.01, 0.001 produce different grayscale structures. As we can see, a larger value of ϵ, say 0.1 or 0.01, may produce a grayscale structure that may still have unsupported edges, as indicated in the red box. The reason behind is that the sum of these gray elements has been less than the prescribed value of ϵ. On the other hand, a too small value of ϵ also finds difficulty in convergence. Thus, the self-supporting constraint is imposed in a soft way following a similar strategy in [33]. Specifically, it decreases from a relatively large value until reaching a target value ϵ, which is set as $1e{-}3$ by default in our study.

Algorithm 1 Main Framework of Self-supporting Topology Optimization

C_1, case of gray elements without self-supporting constraints;
C_2, case of gray elements with self-supporting constraints;
C_3, case of pure 0–1 elements with self-supporting constraints.

1: **Initialize** density $\boldsymbol{\rho} = \boldsymbol{\rho}_0$, case tag $P_{id} = C_1$ // default case without self-supporting constraint
2: **while** $\boldsymbol{\rho}$ not converge **do**
3: *Volume Reduction:* $f = (1-\delta)f$ // f: the volume fraction in Eq. (7); δ : evolution rate set 0.02 by default.
4: *FEM Computing*
5: **if** $P_{id} == C_1$ **and** $M_{nd} < M_{nd}^0$ **then** // M_{nd}^0: expected gray element occupation
6: *Choose print direction*
7: $P_{id} \leftarrow C_2$ // conversion from cases C_1 to C_2
8: **end if**
9: *Sensitivity Analysis according to the value of P_{id}*
10: *Optimization with MMA according to the value of P_{id}*
11: **if** $P_{id} == C_2$ **and** $U(\boldsymbol{\rho}) \le \epsilon$ **then** // ϵ: bound of unsupported element number
12: $P_{id} \leftarrow C_3$ // conversion from cases C_2 to C_3
13: **end if**
14: **if** $P_{id} == C_3$ **then**
15: *Set $\boldsymbol{\rho}$ to 0–1 values*
16: **end if**
17: **end while**



(a) $\epsilon = 0.1$ **(b)** $\epsilon = 0.01$ **(c)** $\epsilon = 0.001$

Fig. 4 Different values of ϵ, as used to describe the self-supporting situation of a structure, produce different ranges of self-supporting elements. Un-self-supported regions are observed in the red boxes

2.2.4 Optimization in a volume reduction loop

The optimization process is embedded in a volume reduction loop, as described in step 3 in **Algorithm 1**. It mainly aims to slowly reduce the target volume so as to provide sufficient design space to impose self-supporting constraint during the optimization process. Specifically, during each optimization step, the target volume fraction is also reduced at a certain evolutionary rate $\delta = 0.02$ by default, or $f = (1-\delta)f$ for the problem in Eq. (7). As can be seen from Fig. 2, using volume reduction may produce a structure of smaller/better compliance in (e) than those produced without volume reduction in (f).

2.3 Framework overview

3 Numerical details

3.1 Sensitivity analysis

The sensitivity of the self-supporting constraint $U(\boldsymbol{\rho})$ can be derived straightforwardly from (7) and given as:

$$\frac{\partial U}{\partial \rho_e} = \begin{cases} 2\rho_e & \text{if } e \in \mathcal{M}_U, \\ 0 & \text{if } e \notin \mathcal{M}_U, \end{cases} \tag{12}$$

where $\mathcal{M}_U$ is the set of unsupported elements.

We comment here that the sensitivity of self-supporting constraint is computed following the idea present in [33]. Devising an improved sensitivity calculation further considering the element range $e \in \mathcal{M}_u$ may help improve the convergence and deserves future research effort.

Derivations of the sensitivities of the objective function $C(\boldsymbol{\rho})$ or of the volume constraint $V(\boldsymbol{\rho})$, involved in (7), are totally the same as those done in previous studies [9,34]. Integrating these sensitivities with thickness control can be achieved using the Heaviside filter [35], as will be demonstrated in Sect. 4. Details are not further explained here.

3.2 Discrete convolution for efficient unsupported element detection

Computing the sensitivity (12) of the self-supporting constraints requires detecting the set $\mathcal{M}_U$ of all the unsupported elements. A novel convolution operator is further devised below to accelerate the detection process.

Given a discrete structure $\mathcal{M}$ of size $N \times M$ in 2D, we can see from (3) that an element $e(n, m) \in \mathcal{M}$ is supported if the summation of the densities of its supporting elements is larger than zero, or specifically,

$$\sum_{n-1 \le r \le n+1} \rho(r, m-1) > 0. \tag{13}$$

The newly introduced *self-supporting convolution operator* is designed based on this observation. Specifically, suppose the overhang angle is 45°. The associated 2D self-supporting kernel matrix $\mathcal{H}$ of size 3×3 is defined in Fig. 5. A new matrix $\boldsymbol{\rho}_S$ is then computed via performing the convolution between the density matrix $\boldsymbol{\rho}$ and $\mathcal{H}$, or

$$\tilde{\boldsymbol{\rho}}_S = \operatorname{sign}(\boldsymbol{\rho} * \mathcal{H}), \tag{14}$$

1	1	1
0	0	0
0	0	0

Fig. 5 2D self-supporting convolution kernel matrix $\mathcal{H}$ to detect supported elements for an overhang angle of 45°

where $\boldsymbol{\rho} * \mathcal{H}$ is the convolution between matrices $\boldsymbol{\rho}$ and $\mathcal{H}$, whose (n, m) element is defined as

$$(\boldsymbol{\rho} * \mathcal{H})(n, m) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} \rho(n-i, m-j) \cdot \mathcal{H}(i+1, j+1), \tag{15}$$

and the sign function

$$\operatorname{sign}(x) = \begin{cases} 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases} \tag{16}$$

Correspondingly, we have the set of supported elements $\mathcal{M}_S$ of the discrete structure $\mathcal{M}$,

$$\mathcal{M}_S = \{e(n, m) \mid \tilde{\boldsymbol{\rho}}_S(n, m) = 1\}, \tag{17}$$

for $\tilde{\boldsymbol{\rho}}_S$ defined in (14).

The above expression assumes a 0–1 distribution of $\boldsymbol{\rho}$, while in the SIMP approach as studied here, the density matrix $\boldsymbol{\rho}$ usually has entry value ranging from 0 to 1. A specific value usually needs to be set to replace 0 in (16) for practical applications.

The basic procedure of the convolution computation is further shown in Fig. 6 and explained below. For each element e under consideration, a 3×3 matrix $\boldsymbol{\rho}_e$ centering at e is selected. This is then followed by its Hadamard product, i.e., the element-by-element product between the matrices,

$$C_e = \tilde{\mathcal{H}} \odot \boldsymbol{\rho}_e, \tag{18}$$

where $\tilde{\mathcal{H}}$ is the rotation of matrix $\mathcal{H}$ at a degree of 180°. The convolution value of element e is the summation of all the values in the derived matrix C_e.

The above procedure works for every element e. For the boundary elements, an additional loop of void elements is added. Note also that the bottom elements are always taken as supported considering the fact that they are always supported by the baseboard of the fabrication device.

Once the set $\mathcal{M}_S$ of supported elements of $\mathcal{M}$ is determined from (17), the set of unsupported elements is derived consequently,

$$\mathcal{M}_U = \mathcal{M} \backslash \mathcal{M}_S. \tag{19}$$

The convolution procedure in 3D is similar to that in 2D, and the corresponding kernel matrix $\mathcal{H}$ is shown in Fig. 7.

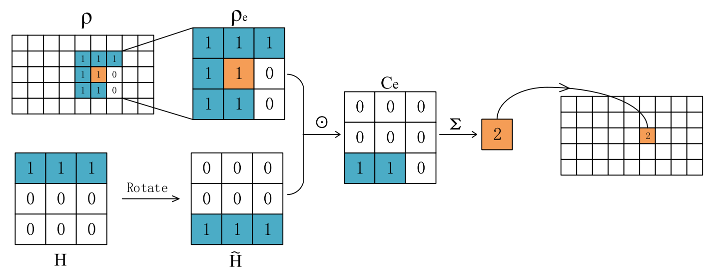


Fig. 6 Discrete convolution procedure in 2D for detecting supported elements using the designed self-supporting Kernel matrix $\mathcal{H}$ in Fig. 5

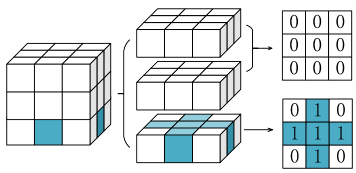
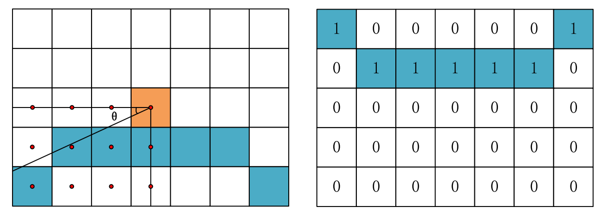


Fig. 7 3D self-supporting convolution kernel matrix $\mathcal{H}$ to detect supported elements for an overhang angle of 45°



(a) Supporting elements **(b)** Convolution Kernel matrix $\mathcal{H}$

Fig. 8 Procedure for building a self-supporting convolution kernel matrix for a general overhang angle θ

Table 1 Time–cost comparison between direct enumeration and using convolution for detecting supported elements in 2D and 3D cases

Domain size	Enumeration (s)	Convolution (s)	Ratio
80 × 40	0.0094	0.0001	94
320 × 80	0.0380	0.00035	106.16
600 × 400	0.5214	0.0034	152.11
40 × 40 × 20	0.1419	0.0010	139.13
100 × 100 × 100	4.7974	0.0324	147.82
500 × 200 × 100	114.7641	0.7605	150.88

Extension of the approach to general overhang angle will be later explained in Sect. 3.3.

In order to demonstrate the effect of using the convolution operator, we compare in Table 1 the computational time in detecting the supported elements using the convolution operator and using direct enumeration element by element. As we can see, almost two orders of speedup are observed from the results. The time of detecting supported elements is not ignorable compared with FE computations. For example, one step of FE for the size of 600 × 400 takes 0.4163 s, and the enumeration takes 0.5214 s. The effect is important for practical applications, particularly on 3D complex structures with millions of elements, considering that FE computations can be implemented in parallel and accelerating the detection of supported elements then becomes prominent.

Note that various convolution operators have been designed and used as filters in topology optimization for design control, for example for removing checkerboard patterns or thickness control [36]. They usually aim to compute an element's density or sensitivity via averaging those of the elements around it. Different from these studies, the convolution is used here for detecting unsupported elements. It does not change the element densities during the optimization process.

3.3 Extension to general overhang angle

Extending the above procedure to a general overhang angle θ is further explained below. The only difference from the case of 45° is the construction of the kernel matrix $\mathcal{H}$ involved in (18).

As shown in Fig. 8a, given an overhang angle θ, a straight line L passing through the center of an element e and with a slope angle θ is drawn. Then, the first element in each column whose centers are below line L is taken as the supporting element with respect to element e. Their densities are set as 1 and others' 0, which together determines a matrix $\bar{\mathcal{H}}$. Rotating $\bar{\mathcal{H}}$ with respect to the center e at a degree of 180° gives the convolution kernel matrix $\mathcal{H}$, as shown in Fig. 8b. The above procedure works for building convolution Kernel using multiple layers; the more layers taken, the more accurate of the built kernel matrix for detecting the supported elements.

4 Examples

Extensive 2D and 3D examples are performed to test the performance of the proposed approach. For illustration purposes, the material, load and geometry data are chosen to be dimensionless. The Young's modulus and Poisson's ratio of the solid material are set as $E = 1$ and $\upsilon = 0.3$ for all examples. The penalization factor is set to a value of 3. The minimal thickness is set to be 2, and the overhang angle is set to be 45° if not explicitly specified. The print direction is selected during each optimization process and marked in the example figure. The 2D examples were implemented in

Table 2 Summary of the numerical results for various tested 2D examples: $\#\mathcal{M}$, element number of the design domain; $\#\mathcal{M}_U$, C_{ref}, number of unsupported elements of structured computed from topology optimization without considering self-supporting constraints and its associated compliance; C, compliance of the self-supporting structure computed using the proposed approach

	$\#\mathcal{M}$	$\#\mathcal{M}_U$	C_{ref}	C	C/C_{ref} (%)
Beam	9000	24	92.7	92.8	100.11
Beam (hole)	7755	25	115.4	117.4	101.73
Beam ($r = 1.5$)	9000	24	92.7	92.8	100.11
Beam ($r = 2$)	9000	18	92.4	92.8	101.43
Beam ($r = 3$)	9000	12	92.6	92.9	100.32
Beam (concentrated)	14,400	21	322.9	323.0	100.31
Beam (distributed)	14,400	12	255.4	256.1	100.27
Beam (mixed)	14,400	60	31971.1	32514.3	101.70
Beam (vf = 0.6)	9000	24	92.7	92.8	100.11
Beam (vf = 0.5)	9000	97	105.8	106.8	100.95
Beam (vf = 0.4)	9000	37	127.4	128.3	100.71
Beam (vf = 0.25)	9000	55	196.8	202.9	103.10
Beam (Angle = 30)	9000	12	105.8	113.6	107.37
Beam (Angle = 45)	9000	97	105.8	106.8	100.95
Beam (Angle = 60)	9000	425	105.8	141.3	133.55
MBB	38,400	1874	185.7	191.3	103.02
Square	22,500	474	1312.8	1544.7	117.66

The Cantilever beam examples are described in Figs. 9 and 17, the MBB example in Fig. 19, the 2D squares in Fig. 21, the 3D examples in Figs. 22, 25 and 26

MATLAB, and the 3D examples were implemented in C++ and GPU for parallel computations on a computer of 3.2G CPU, 8.0G RAM and GeForce GTX 970 Graphic card.

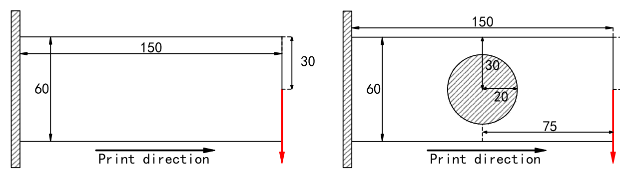
The examples include the classical cantilever beam, MBB, a 2D square example and three 3D examples. The cantilever beam is used to illustrate various aspects of the approach: basic performance on a rectangular domain or a general domain, iteration process, thickness control, different volume fractions, different overhang angles, different types of external forces. The MBB shows the approach's ability in handling constraints of multiple print directions. The 2D square example demonstrates the approach's performance in the case of complex topological structure obtained at distributed external forces. The 3D examples are further used to demonstrate the approach's ability in handling complex 3D models of millions of DOFs via parallel implementation.

Following previous studies [13,14,20,21], we measure the ability of a self-supporting topology optimization approach in maintaining the structure's physical performance using the compliance ratio

$$\frac{C}{C_{\text{ref}}}, \tag{20}$$

where C_{ref}, C is, respectively, the compliance of the structure computed with or without considering self-supporting constraint.

The computational results for 2D examples are first summarized in Table 2; cases of 3D examples are explained later.



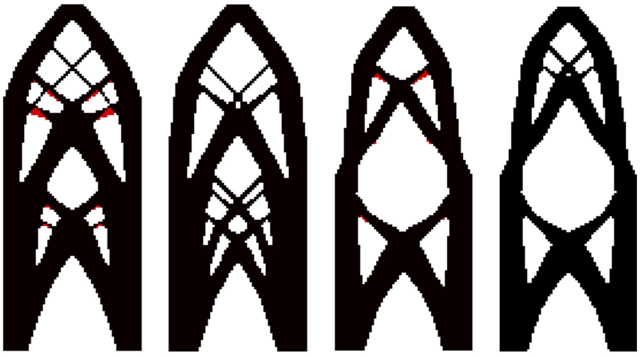
(a) Rectangular domain **(b)** General domain

Fig. 9 Cantilever beam examples with a minimum thickness of 1.5

4.1 Cantilever beam example

The cantilever beam, as shown in Fig. 9, is first tested. The model on the top has a 150 × 60 rectangular domain and has a target volume fraction of 0.6. The model on the bottom has general domain made via cutting a circular hole within the left one and a target volume fraction of 0.5. Both models are fixed on the left edge with an external force exerted on the middle point of its right edge. The print direction is determined from left to right.

Without considering the self-supporting structure, the structure in Fig. 10a, c is obtained where the elements in red are those that cannot be successfully printed out. The proposed self-supporting topology optimization approach results in the structures in (b) and (d), both of which do not contain any unsupported elements. We can see from the results that the range containing unsupported elements in (a),(c) moves upward in (b),(d) to adapt the requirement of self-supporting. In addition, it is also very interesting to notice that several local parts of (b) or (d) are very differ-



(a) Support-needed **(b)** Self-supporting **(c)** Support-needed **(d)** Self-supporting

Fig. 10 Computational results for the cantilever beam example in Fig. 9 at a volume fraction of 0.6. See also Table 2 for more details

ent from those of (a) or (c) to satisfy the self-supporting constraint and for the structures' maximal physical performance, although their overall structures are simultaneously maintained. The structures computed with or without self-supporting constraints have a very close compliance, with a compliance ratio, respectively, of 100.11% and 101.73% for the left and right examples.

In handling the bottom model of a general design domain, we work on the rectangular domain following the procedure below. In each step of the optimization iteration, the density of each element within the circular domain is set as 0, and then the convolution operation (detailed in Sect. 3.2) is performed in the whole rectangular domain to detect the unsupported elements. The above two steps are repeated until convergence.

4.1.1 Iteration performance

The iteration process of the example given in Fig. 9a is further explored by examining the variations in the structure's topology, the number of unsupported elements and the compliance and volume fraction of the derived structure, as shown in Figs. 11 and 13.

The iteration process is divided into the following main steps (see also Fig. 11). Firstly, a topology optimization step without considering the self-supporting constraint, that is, case C_1 in Algorithm 1, is performed, and results in the "gray" structure in (a). After this, the relaxed self-supporting constraint is added in the optimization iteration step (case C_2 in Algorithm 1), producing a structure in (b). The derived "gray" structure is then transformed into a black–white structure using the Heaviside project filter, as given in (c). After this, the self-supporting topology optimization process (case C_3 in Algorithm 1) is iterated to reduce the number of unsupported elements while simultaneously optimizing its physical performance and maintaining its volume fraction, producing

(a) step 20 **(b)** step 44 **(c)** step 45 (22)

(d) step 54 (12) **(e)** step 82 (6) **(f)** step 101 (3) **(g)** step 166 (0)

Fig. 11 Keyframe figures at different iteration steps, where the number in brackets stands for the number of unsupported elements

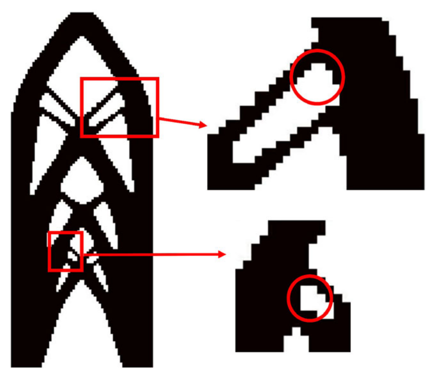
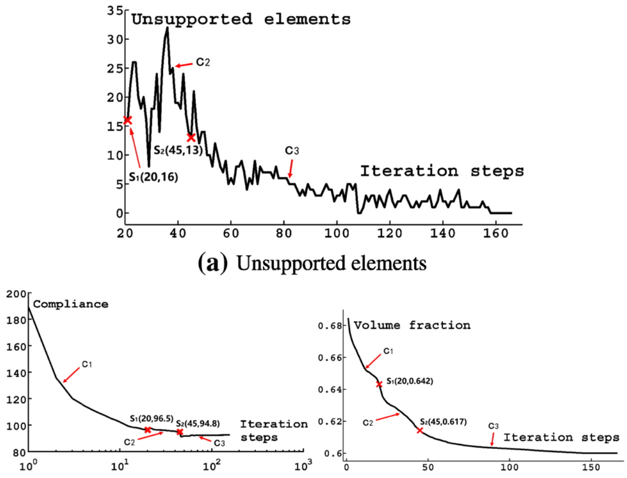


Fig. 12 Close-up of the structure in Fig. 11g, where all the elements are self-supported. Note particularly that the top two elements in the red circle are self-supported by the left-bottom and right-bottom elements below them in the next layer

the structures in (d), (e), (f) and ultimately the final structure in (g). The unsupported elements are marked red in Fig. 11c–g and illustrated in the caption. Their number is gradually decreased during the optimization iteration process. Figure 12 also shows a close-up of the final optimized structure in Fig. 11g, where all the elements are self-supported.

Figure 13 shows the overall performance of the approach in the different cases of C_1, C_2, C_3 and the associated keyframe labeled S_1, S_2, in terms of the variation in the



(a) Unsupported elements
(b) Compliance
(c) Volume fraction

Fig. 13 Variations in the structure's number of unsupported elements, compliance and volume fraction during the optimization iteration steps. The critical situations marked S_1, S_2 indicate two keyframes: conversion from cases C_1 to C_2 and conversion from cases C_2 to C_3. C_1, case of gray elements without self-supporting constraints; C_2, case of gray elements with self-supporting constraints; C_3, case of pure 0–1 elements with self-supporting constraints. The three cases are embedded in an iterative volume reduction procedure in Algorithm 1

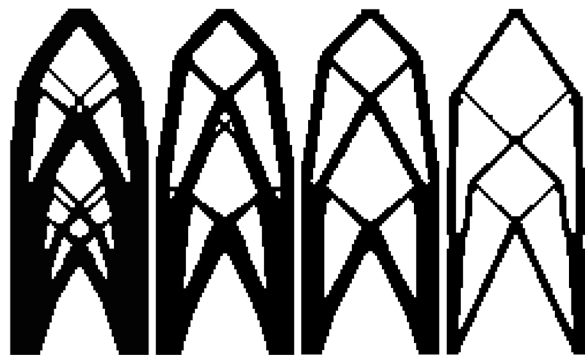
number of unsupported elements, the target compliance and the volume fraction. As can be seen, as the iteration step increases, the number of unsupported elements decreases until finally reaching zero. However, fluctuations in the number of unsupported elements may happen during the iteration process. The structures' compliance and volume fraction decrease and finally reach a stable state. Note here different from classical SIMP, where the volume fraction is maintained, an iterative volume reduction strategy is applied in Algorithm 1, as can be observed in Fig. 13c.

4.1.2 Thickness control

The proposed approach is also able to control the structure's thickness to meet different device requirements, with a direct additional usage of a density filter for thickness control. Figure 14 shows the obtained self-supporting structures, respectively, of thicknesses 1.5, 2 and 3. The overall structures of these three self-supporting structures are similar, and as the minimum thickness increases, the slender beams are removed gradually. Consequently, the smaller the thickness required, the more the details preserved in the final structures. The associated compliances of the three structures are very close to each other, as summarized in Table 2, of ratios to their reference structures, respectively, $C/C_{\text{ref}} = 100.11\%, 101.43\%, 100.32\%$.

(a) $r_{min} = 1.5$ **(b)** $r_{min} = 2$ **(c)** $r_{min} = 3$

Fig. 14 Numerical results for the Cantilever beam example in Fig. 9a at different thicknesses of 1.5, 2 and 3. See also Table 2 for more computational details. The target volume fraction is set 0.6



(a) 0.6 **(b)** 0.5 **(c)** 0.4 **(d)** 0.25

Fig. 15 Self-supporting structures computed using the proposed approach at different volume fractions. See also Table 2 for more computational details

4.1.3 Different volume fractions

Performance of the approach is also tested at constraints of different volume fractions, and the computed structures are shown in Fig. 15. Such self-supporting structures become harder to obtain for small value of volume fractions. As can be observed from the results, the self-supporting constraints can still be satisfied although the number of elements decreases as the volume fraction becomes smaller.

4.1.4 Different overhang angles

As have been explained previously in Sect. 3.3, the proposed approach can also work for overhang angle different from 45° via using different convolution Kernel matrices $\mathcal{H}$. We demonstrate its performance still using the cantilever beam example in Fig. 9a at a volume fraction of 0.5, for three different overhang angles: 30°, 45°, 60°. The associated convolution kernel matrices $\mathcal{H}$ for angles of 30°, 60° are also shown in Fig. 16d, e. As can be observed from

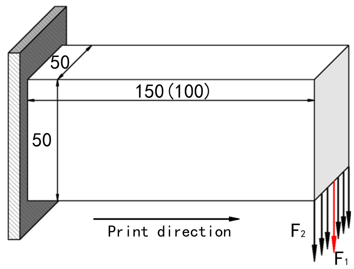


Fig. 16 Self-supporting structures obtained using the proposed approach at different overhang angles at a volume fraction of 0.5, and their associated convolution kernel matrices. See also Table 2 for more computational details

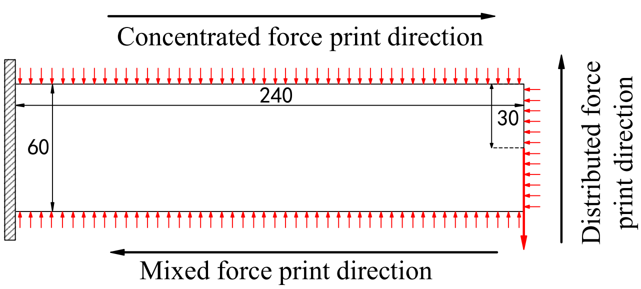


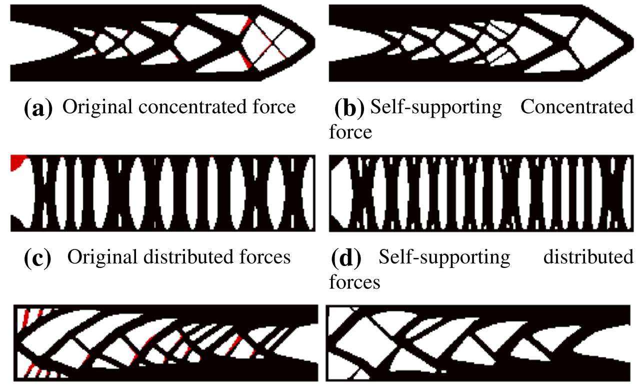
Fig. 17 Cantilever beam example with distributed external forces

the examples, as the overhang angle becomes bigger, the boundary edges moves upward to meet the self-supporting constraints. The structure's minimal compliance is still well kept at these different angles, with C/C_{ref}, respectively, of 107.37%,100.95% and 133.55%. It is also noticed that the larger overhang angle deteriorates the structures' physical performance. Similar phenomenon was also observed in previous studies [13,14,21].

4.2 2D cantilever beam example with different types of forces

In order to test the ability of the proposed approach in selecting the print direction and its performance in finding an optimized structure at different external loadings, the 2D cantilever beam example in Fig. 17 is tested under different types of forces, respectively, of concentrated force, distributed force and mixed forces. In this example, the design domain is discretized into 240 × 60 square FE elements. The volume fraction is 0.6 and the minimum thickness is 2. The concentrated force is exerted on the middle point of the right edge and points downward. The distributed force is exerted evenly on the bottom, top and right edges of the model, while the case of mixed forces takes both into account.

For each of the three cases, different print directions are chosen and shown in Fig. 17 by the proposed approach. The corresponding optimized structures are also shown in Fig. 18. As can be observed from the results and the summary in Table 2, different boundary conditions may require different print directions and produce different optimized structures, which all can be handled successfully via the proposed approach. Compliance of the reference structure is maintained at a compliance ratio of 100.31%, 100.27% and 101.70%.



(a) Original concentrated force **(b)** Self-supporting Concentrated force **(c)** Original distributed forces **(d)** Self-supporting distributed forces **(e)** Original concentrated and distributed forces **(f)** Self-supporting concentrated and distributed forces

Fig. 18 Derived optimized structures of the cantilever beam example in Fig. 17 under different types of external forces. In all cases, structure similarity and compliance closeness are observed between the structures with or without considering the self-supporting constraints. Different print directions were chosen for these different cases for ease of convergence and optimized physical performance. See also Table 2 for more computational details

4.3 2D MBB example constrained by more than one print direction

The proposed approach is also able to simultaneously take into account more than one print direction constraints, provided they do not conflict with each other. This is illustrated using the classical MBB problem in Fig. 19. Due to the symmetry of the model's structure and boundary conditions, only half of the computational domain is used here which consists of 160 × 30 square FE mesh elements. The volume fraction is 0.5 and the minimum thickness is 1.5.

The aim is to produce a self-supporting structure maintaining the mirror symmetry of the original model. Thus, the self-supporting requirement has to be added in both directions: from right to left and from left to right for the half-sized structure in Fig. 19b. As a result, a self-supporting structure in both directions is obtained in Fig. 20b, as compared with the support-needed structure in Fig. 20a, where the unsupported elements are plotted in red. The compliance of the support-needed structure and the self-supporting structure is, respectively, 185.7 and 191.3, at a relative ratio of 103.02%.

Note that the 2D MBB problem of the same domain size was also tested by Gaynor and Guest in [21], where the print

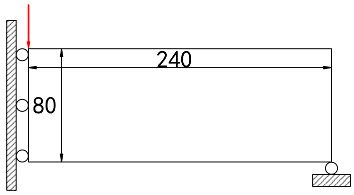


Fig. 19 2D MBB example: Due to its symmetry, only one half of the model needs to be studied in the topology optimization process

direction was manually set from bottom to top. Such setting thus does not require constraints of multiple constraints.

4.4 Complex internal structure at distributed force

The proposed approach is also able to produce self-supporting structure for complex internal structure, as demonstrated using the square example at distributed forces in Fig. 21a. The computed support-needed structure and self-supporting structure are, respectively, given in Fig. 21b, c. It can be seen from the results that the original support-needed structure has many small flat edges which prevent the structure to be fabricated without a large number of additional supports. Such unsupported elements have disappeared in the optimized self-supporting structure in Fig. 21b, despite the structure's high complexity. In addition, the self-supporting structure also has a close compliance with that of the original support-needed structure, respectively, of 1544.7 and 1312.8 at a relative compliance ratio of 117.66%. A self-supporting requirement is very necessary for such highly complex structure, as computing the supports or their removal would be extremely troublesome if not impossible.

4.5 3D Examples

The proposed approach has also been implemented for complex 3D examples of high DOFs, including the classical benchmark examples: a 3D wheel, a 3D cantilever and a newly devised examples of a 3D desk. The domain size and their associated computational time are summarized in Table 3.

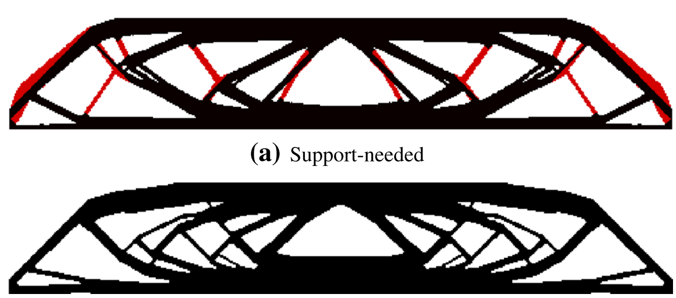
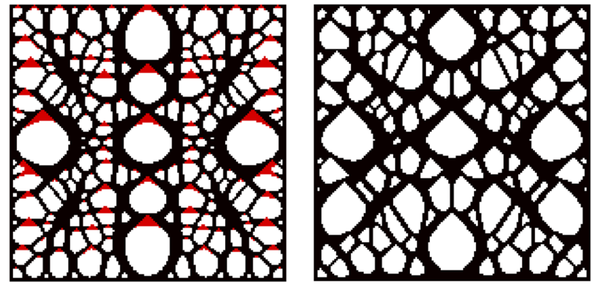


Fig. 20 Topology optimization result of the MBB model without **a** or with **b** self-supporting structure. The self-supporting constraint is needed simultaneously in two different directions: from left to right and from right to left, so that the mirror symmetry of the original structure can be maintained. See also Table 2 for more computational details



(a) Support-needed **(b)** Self-supporting

Fig. 21 Problem of a square under a distributed force, and its computational results. See also Table 2 for more computational details

4.5.1 3D Wheel

The 3D wheel example in Fig. 22 consists of $100 \times 100 \times 100$ cubic mesh elements. The four corners of the bottom face are fixed, and a concentrated force is exerted on the middle point of the bottom face. The target volume fraction is 0.25 and the minimum thickness is 2. The print direction is chosen as from top to bottom.

The computed self-supporting structure using the proposed approach is shown in Fig. 23b, as compared with its counterpart of support-needed structure in Fig. 23a. The corresponding structure slices at $x = 25, 35, 45$ of both the self-supporting and support-needed structures are also shown and compared in Fig. 23f–h, and c–e. As can be seen from the results, the originally flat regions of the support-needed structure, which cannot be fabricated without supports, have been optimized to meet the self-supporting requirement. The resulting structure is totally self-supporting for direct fabrication purpose, and its compliance is 87.32, very close to that of the original support-needed one of 86.16, at a relative ratio of 1.01%.

4.5.2 3D cantilever

Two different 3D cantilever examples are tested here as illustrated in Fig. 25: one of size $100 \times 50 \times 50$ exerted by point loadings, and another one of size $150 \times 50 \times 50$ exerted by edge loadings; a same example to the latter was also studied

(a) Support-needed

(b) Self-supporting

Table 3 Time comparison for 3D examples

Example	Wheel in Fig. 22	Cantilever (a) in Fig. 22	Cantilever (b) in Fig. 22	Desk in Fig. 26 (a)
Size	100 × 100 × 100	100 × 50 × 50	150 × 50 × 50	120 × 120 × 80
Time	35 m	11 m	17 m	66 m

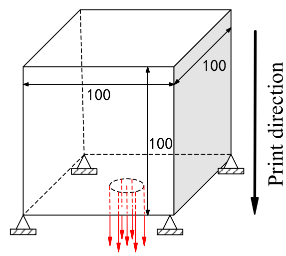


Fig. 22 The wheel example

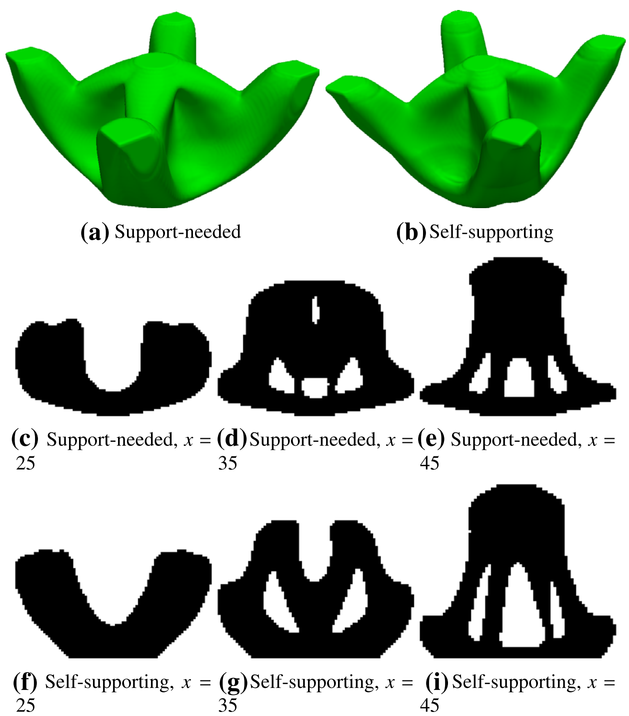


Fig. 23 Self-supporting structure generated using the proposed approach and its comparison with that of the support-needed one generated without the consideration of self-supporting constraint. Both the 3D structure and their slices at $x = 25, 35, 45$ are shown

in [14]. The target volume fraction is 0.3 and the minimum thickness is 2. The print direction is chosen as from top to bottom. The computed self-supporting structures are shown in Fig. 25, which is directly to be fabricated without any additional support materials.

4.5.3 3D Desk

A more complex 3D desk problem(also called hanging bridge), as shown in Fig. 26a, is designed to further test performance of the proposed approach. The example is of size

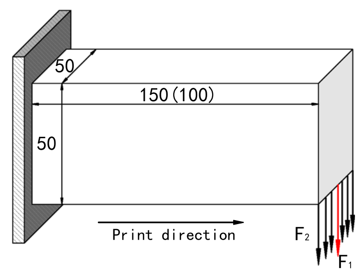


Fig. 24 A 3D cantilever example of size $100 \times 50 \times 50$ at a point loading F_1 and of size $150 \times 50 \times 50$ at an edge loading F_2

$120 \times 120 \times 80$ and consists of 1.152 millions of elements. In this example, the four bottom corners are fixed and the top face is exerted by a uniformly distributed force pointing downward. The target volume fraction is 0.3, the minimum thickness is 2 and the print direction is chosen from right to left. The final generated self-supporting structure is shown in Fig. 26b together with slices in (c) and (d), taking 66 min. It is also interesting to note that the four legs of the desk are not totally solid but take porous bone-like structures to balance the constraint of the object weight and the target compliance.

4.6 Summary

As can be observed from these examples, using the proposed self-supporting topology optimization framework, the edges or faces of the original support-needed structure are aligned toward the print direction so that all elements can be successfully fabricated. In addition, the produced structures with or without self-supporting constraints have very close shapes, and their compliance difference is maintained within a very small or negligible range. The performance demonstrates the strength of the proposed framework in designing self-supporting structures and simultaneously maintaining their optimized structures and physical performances.

5 Limitations and future work

A novel self-supporting topology optimization framework is developed in this paper for applications in additive manufacturing. The usage of convolution operator and the associated numerical techniques enables the self-supporting structure to be generated efficiently. In the framework, special techniques are developed to fine-tune the convergence of the approach, as described in Sect. 2.3, so that an optimized structure of

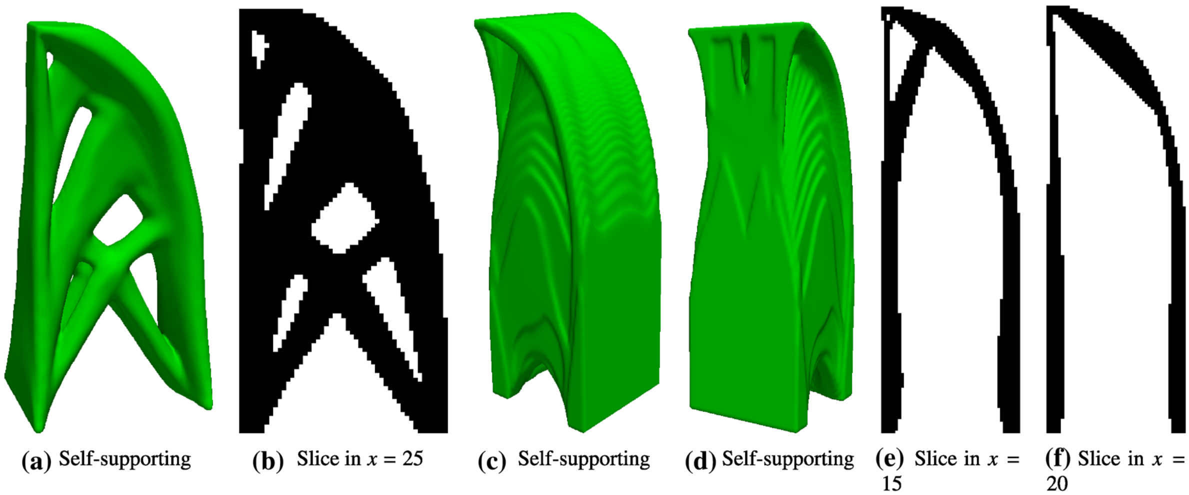


Fig. 25 Generated self-supporting structures, together with its slices, for two difference cases: **a**, **b** for structure of size $100 \times 50 \times 50$ at a point loading F_1, and **c**–**f** of size $150 \times 50 \times 50$ at an edge loading F_2 for the example in Fig. 24

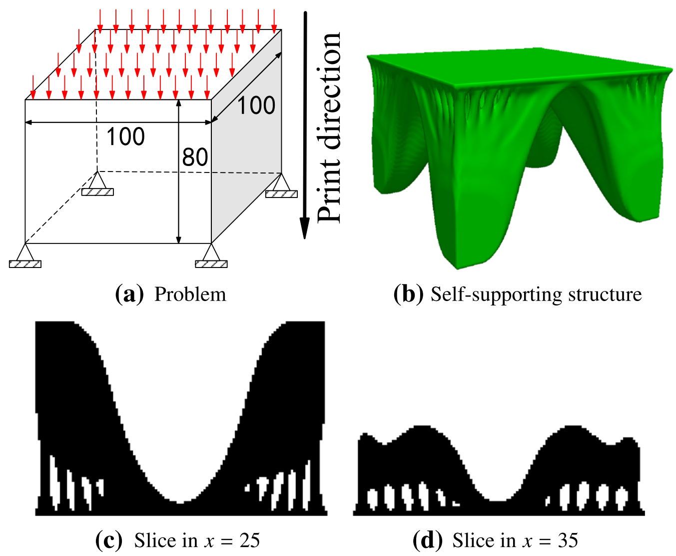


Fig. 26 A 3D desk example, its generated self-supporting structure and the associated slices at $x = 23$ and $x = 35$

close compliance and shape to the benchmark structure can be ultimately generated. Using the technique helps to produce a reliable structure, but it, on the other hand, also involves issues on choosing appropriate parameter values, and further research effort is to be devoted to resolving the issue.

The framework chooses an appropriate print direction before taking into account the self-supporting constraint during the optimization process. If ignoring the step of choosing the print angle and setting an arbitrary print direction, the approach may fail to produce a converged structure. For example, an optimized self-supporting structure is hard to obtain for the MBB example in Fig. 19 at a different direction. Such phenomenon may not prevent the approach in generating a self-supporting structure suitable for additive manufacturing, but may hinder its usage in specific applications. We also note that recent work of [13,22,32] does not involve such a process of selecting the print angle.

The framework is at present implemented using regular square or cubic elements, which are dominated in researches of topology optimization. On the other hand, the overall framework also works for general domains consisting of irregular quad- or hex-elements, but is also limited by the fact that the convolution operator presented in Sect. 3.2 is no longer applicable as they become different for different elements. Thus, the element-by-element enumeration has to be taken and will reduce the computational efficiency.

The framework can also be extended to porous interior designs of 3D free-form structures that do not need any additional supports within its interior. Such supports would otherwise be very difficult to remove. In addition, besides the self-supporting requirements studied here, other fabrication constraints, such as hanging bridge, also need to be included so that ultimately an optimized structure ready to be fabricated can be directly generated.

Acknowledgements The valuable comments and suggestions from the anonymous reviewers are greatly appreciated. The work described in this paper is partially supported by the NSF of China (No. 61872320) and the National Key Research and Development Program (No. 2016YFC1101302) from the MIST of China.

References

1. Bendsoe, M., Kikuchi, N.: Generating optimal topologies in structural design using a homogenization method. Comput. Methods Appl. Mech. Eng. **71**(2), 197–224 (1988)
2. Bendsøe, M.P.: Optimal shape design as a material distribution problem. Struct. Optim. **1**(4), 193–202 (1989)
3. Xie, Y.M., Steven, G.P.: A simple evolutionary procedure for structural optimization. Comput. Struct. **49**(5), 885–896 (1993)
4. Huang, X., Xie, Y.M.: A further review of ESO type methods for topology optimization. Struct. Multidiscip. Optim. **41**, 671–683 (2010)
5. Wang, M.Y., Wang, X., Guo, D.: A level set method for structural topology optimization. Comput. Methods Appl. Mech. Eng. **192**, 227–246 (2003)
6. van Dijk, N.P., Maute, K., Langelaar, M., van Keulen, F.: Level-set methods for structural topology optimization: a review. Struct. Multidiscip. Optim. **48**(3), 437–472 (2013)
7. Qian, X.: Topology optimization in B-spline space. Comput. Methods Appl. Mech. Eng. **265**, 15–35 (2013)
8. Hughes, T.J.R., Cottrell, J.A., Bazilevs, Y.: Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput. Methods Appl. Mech. Eng. **194**(39), 4135–4195 (2005)
9. Sigmund, O., Maute, K.: Topology optimization approaches: a comparative review. Struct. Multidiscip. Optim. **48**(6), 1031–1055 (2013)
10. Sigmund, O.: On Topology Optimization with Manufacturing Constraints. Springer, Netherlands (2006)
11. Serphos MR (2014) Incorporating AM-specific manufacturing constraints into topology optimization. Master thesis. Report no: EM 2014.011, Delft University of Technology. https://repository.tudelft.nl/islandora/object/uuid:64200c38-704d-426f-a423-d0dd6f543213?collection=education
12. Liu, J., Ma, Y.: A survey of manufacturing oriented topology optimization methods. Adv. Eng. Softw. **100**, 161–175 (2016)
13. Langelaar, M.: An additive manufacturing filter for topology optimization of print-ready designs. Struct. Multidiscip. Optim. **55**, 1–13 (2016)
14. Langelaar, M.: Topology optimization of 3D self-supporting structures for additive manufacturing. Addit. Manuf. **12**, 60–70 (2016)
15. Daniel, T.: The development of design rules for selective laser melting. University of Wales, Cardiff (2009)
16. Majhi, J., Janardan, R., Schwerdt, J., Smid, M., Gupta, P.: Minimizing support structures and trapped area in two-dimensional layered manufacturing. Computat. Geom. **12**(3–4), 241–267 (1999)
17. Wang, W., Wang, T.Y., Yang, Z., Liu, L., Tong, X., Tong, W., Deng, J., Chen, F., Liu, X.: Cost-effective printing of 3D objects with skin-frame structures. ACM Trans. Graph. **32**(6), 1–10 (2013)
18. Dumas, J., Hergel, J., Lefebvre, S.: Bridging the gap: automated steady scaffoldings for 3D printing. ACM Trans. Graph. **33**(4), 1–10 (2014)
19. Brackett, D., Ashcroft, I., Hague, R.: Topology optimization for additive manufacturing. In: Proceedings of the Solid Freeform Fabrication Symposium, Austin, TX, 2011, pp. 348–362
20. Gaynor, A.T., Guest, J.K.: Topology optimization for additive manufacturing: considering maximum overhang constraint. In: AIAA AVIATION 2014—15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, pp. 1–9 (2014)
21. Gaynor, A.T., Guest, J.K.: Topology optimization considering overhang constraints: eliminating sacrificial support material in additive manufacturing through design. Struct. Multidiscip. Optim. **54**(5), 1157–1172 (2016)
22. Qian, X.: Undercut and overhang angle control in topology optimization: a density gradient based integral approach. Int. J. Numer. Methods Eng. **111**(3), 247–272 (2017)
23. Garaigordobil, A., Ansola, R., Santamaría, J., de Bustos, I.F.: A new overhang constraint for topology optimization of self-supporting structures in additive manufacturing. Struct. Multidiscip. Optim. **58**(5), 2003–2017 (2018)
24. Allaire, G., Dapogny, C., Estevez, R., Faure, A., Michailidis, G.: Structural optimization under overhang constraints imposed by additive manufacturing technologies. J. Comput. Phys. **351**, 295–328 (2017)
25. Mirzendehdel, A.M., Suresh, K.: Support structure constrained topology optimization for additive manufacturing. Comput. Aided Des. **81**, 1–13 (2016)
26. Lee, M., Fang, Q., Cho, Y., Ryu, J., Liu, L., Kim, D.-S.: Support-free hollowing for 3D printing via voronoi diagram of ellipses. Comput. Aided Des. **101**, 23–36 (2018)
27. Wang, W., Liu, Y.-J., Wu, J., Tian, S., Wang, C.C., Liu, L., Liu, X.: Support-free hollowing. IEEE Trans. Visual Comput. Graph. **24**(10), 2787–2798 (2017)
28. Xie, Y., Chen, X.: Support-free interior carving for 3D printing. Vis. Inform. **1**(1), 9–15 (2017)
29. Wu, J., Wang, C.L., Zhang, X., Westermann, R.: Self-supporting rhombic infill structures for additive manufacturing. Comput. Aided Des. **80**, 32–42 (2016)
30. Wu, J.: Continuous optimization of adaptive quadtree structures. Comput. Aided Des. **102**, 72–82 (2017)
31. Chengkai, D., Chenming, W.C.C.L.W., Sylvain, L., Guoxin, F., Yong-Jin, L.: Support-free volume printing by multi-axis motion. ACM Trans. Graph. **37**(4), 1–14 (2018)
32. Langelaar, M.: Combined optimization of part topology, support structure layout and build orientation for additive manufacturing. Struct. Multidiscip. Optim. **57**(5), 1–20 (2018)
33. Guo, X., Zhang, W., Zhong, W.: Explicit feature control in structural topology optimization via level set method. Comput. Methods Appl. Mech. Eng. **272**, 354–378 (2014)
34. Sigmund, O.: A 99 line topology optimization code written in matlab. Struct. Multidiscip. Optim. **21**(2), 120–127 (2001)
35. Guest, J., Prevost, J., Belytschko, T.: Achieving minimum length scale in topology optimization using nodal design variables and projection functions. Int. J. Numer. Methods Eng. **61**(2), 238–254 (2004)
36. Dłaz, A., Sigmund, O.: Checkerboard patterns in layout optimization. Struct. Multidiscip. Optim. **10**(1), 40–45 (1995)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Dengyang Zhao is currently a PhD student at State Key Laboratory of CAD&CG, Zhejiang University. He holds the bachelors degree in Chongqing University (2015). His research interests include topology optimization, additive manufacturing and GPU computing.

Ming Li associate professor at the State Key Laboratory of CAD&CG, Zhejiang University. Dr. Li obtained his PhD from the Chinese Academy of Sciences in 2004. From that on, he worked in Cardiff University, UK and Drexel University, US from 2004 to 2008. His main research topics include: CAD, intelligent design, digital material. Dr. Li has chaired or attended more than 10 national research grants, and published more than 40 international papers on related topics.

Yusheng Liu is currently a professor in State Key Lab. of CAD&CG, Zhejiang University, P.R. China. He received his PhD degree in Mechanical Engineering from Zhejiang Univ., P.R.China in 2000. After that, he continued his postdoctoral research in State Key Lab. of CAD&CG, Zhejiang Univ. and City Univ. of Hongkong in 2000–2003. He was a visiting scholar in Georgia Institute of Technology (GaTech) during 2009–2010. His main research interests are model based systems engineering (MBSE), CAX, content based 3D model retrieval and reuse.