Prevention of enclosed voids in topology optimization using a cumulative sum flood fill algorithm

van der Zwet, Joran; Delissen, Arnoud; Langelaar, Matthijs

DOI
10.1016/j.advengsoft.2023.103530

Publication date
2023

Document Version
Final published version

Published in
Advances in Engineering Software

Citation (APA)
van der Zwet, J., Delissen, A., & Langelaar, M. (2023). Prevention of enclosed voids in topology optimization using a cumulative sum flood fill algorithm. *Advances in Engineering Software*, *186*, Article 103530. https://doi.org/10.1016/j.advengsoft.2023.103530

Advances in Engineering Software 186 (2023) 103530

Contents lists available at ScienceDirect

Advances in Engineering Software

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

Prevention of enclosed voids in topology optimization using a cumulative sum flood fill algorithm

Joran van der Zwet *, Arnoud Delissen, Matthijs Langelaar

Delft University of Technology, Netherlands

ARTICLE INFO

Keywords:
Topology optimization
Enclosed void
Additive manufacturing
Connectivity
Powder bed fusion

ABSTRACT

Topology optimization has seen increased interest with the rise of additive manufacturing (AM) as a fabrication method, because of its ability to exploit the geometric complexity that AM offers. However, AM still imposes some geometric restrictions on the design, most notably on minimum feature size, overhang angles, and enclosed voids. Enclosed voids are problematic because for many AM methods it is impossible to remove supports, unmelted powder or uncured liquid from them. This paper introduces a filter based on a cumulative sum flood fill algorithm to alleviate this issue in a flexible manner. This filter produces a density field where every enclosed void element is rendered solid. It successfully eliminates enclosed voids in both 2D and 3D problems, with low computational cost due to its geometric nature. In addition we demonstrate direct control over the location, amount, and size of powder removal features by varying boundary conditions for the filter, running additional flood fills, and adding morphology operators, respectively.

1. Introduction

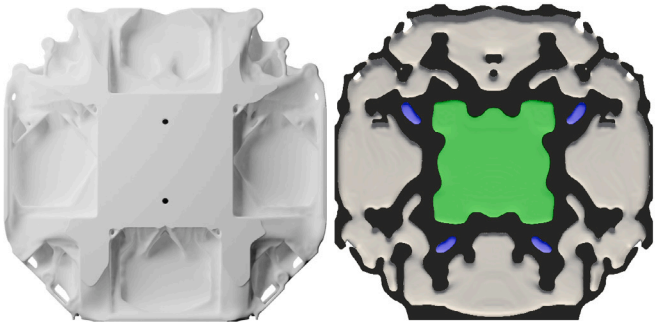
Topology optimization (TO) is a computational design technique which aims to find the optimal material distribution for a given structural problem without requiring a predetermined design concept. Ensuring TO inherently conforms to the shape and feature restrictions that a chosen manufacturing technique entails is an active area of research, as this eliminates the need for an additional post-processing step to make the generated designs manufacturable. This post-processing step requires additional effort once the part has been made, and typically also reduces performance of optimized designs. Numerous contributions have already been made for traditional manufacturing techniques, e.g. casting and milling, to ensure castability e.g. [1–3] and machinability e.g. [4–7], but these are too restrictive when applied to additive manufacturing (AM). TO has seen increased interest in recent years with the rise of AM as a viable manufacturing technique. This is mainly due to the ability of TO to exploit the increased geometric complexity that AM offers, where the notable remaining geometric manufacturing constraints concern overhanging features, minimum size of features and enclosed voids. The former of these constraints have already been studied extensively, therefore the focus of this paper is on the latter.

Enclosed voids prevent access to sacrificial support structures, making it impossible to remove them. In powder- or liquid-based AM, enclosed voids will trap unmelted powder or uncured liquid even when the part is fully self-supporting. Unreachable supports or trapped matter add unnecessary mass, e.g. Reddy et al. [8] found that 82 cm^3 of loose powder was trapped in a topology optimized part of 175 cm^3, which negatively influences the performance of the part. Another example is the optimized part by Delissen et al. [9] given in Fig. 1. In this design the large enclosed void marked with green was made accessible by inclusion of evacuation channels, while the smaller blue enclosed voids were filled. Workarounds such as adding evacuation channels add additional design modification and post-processing steps, therefore a method to suppress enclosed voids during the TO process is desired.

Various studies have already targeted this problem in TO. These can be subdivided into physics-based, geometrical and overhang avoidance categories. From physics-based methods the earliest contribution is the virtual temperature method proposed by Liu et al. [10]. This method replaces any voids with a virtual heat source of high conductivity, while solids are replaced with virtual thermally insulating material. These new material properties are in turn combined with heat dissipation boundaries at the edge and a maximum temperature constraint, which can then only be satisfied by eliminating the enclosed voids or creating channels of void material between the enclosed voids and the dissipation boundaries. The virtual temperature method has been extended further by Li et al. [11] with the addition of different heat dissipation boundaries and additional 3D problems. Luo et al. [12] changed the formulation of the virtual temperature method by making the virtual heat source temperature dependent to better predict the resulting maximum virtual temperature. Wang et al. [13] utilized a similar strategy, based

* Corresponding author.
E-mail address: J.M.vanderZwet-1@tudelft.nl (J. van der Zwet).

https://doi.org/10.1016/j.advengsoft.2023.103530
Received 31 March 2023; Received in revised form 11 July 2023; Accepted 20 August 2023
Available online 22 September 2023



(a) Top view. (b) Cross section, with enclosed voids marked by green and blue.

Fig. 1. Optimized part with enclosed voids.
Source: Image taken from [9].

on an electrostatic model rather than heat conduction. Yamada and Noguchi [14] applied a virtual temperature enclosed void constraint to level-set based topology optimization. A diffusion physics approach is taken by Sabiston and Kim [15], where a constraint is formulated based on the time a virtual particle takes to reach the domain boundary. Finally, in the approach proposed by Donoso et al. [16] an eigenvalue problem is solved to ensure that all voids are connected to each other, which was expanded further in [17]. Physics-based approaches allow for the control over whether or not enclosed voids are allowed to exist, but offer less explicit control over the amount and size of these void features, so parameters need to be chosen carefully or there is a risk of heavily restricting the possible geometry. They also require solving an additional physical field, adding significantly to the computational effort required.

A geometrical approach is taken by Zhou et al. [18] in the side constraint method, in which the optimization is carried out by changing the size and shape of predetermined void features. These void features have their midpoint restricted to the outside boundary, ensuring that any void is always accessible from this boundary. Gaynor et al. [19] use a similar approach in the void projection method, which also changes the design problem so that the optimization is carried out on where the voids exist and subsequently restricting these to nucleate from a predetermined surface. The structural connectivity control approach proposed by Xiong et al. [20] is based on dividing void elements into different sets and using a shortest path algorithm to ensure all void sets are connected to the boundary. Liu et al. [21] employ a similar approach combined with a genetic algorithm to find the optimal path between voids. Some geometrical approaches encounter the same disadvantage as physics-based approaches, in that the geometry can be unnecessarily restricted by the imposed connectivity constraint. Other geometrical approaches utilize fully non-differentiable operations, which are more applicable as a post processing step.

The third category consists of overhang avoidance methods, which focus on ensuring that any enclosed void is self-supporting so that there is no unreachable support material left in enclosed voids. Luo et al. [12] used their altered virtual temperature method to generate self-supporting structures in an enclosed void, which was further expanded by Luo et al. [22] to generate porous support material in any enclosed void. Wang [23] also applied a virtual temperature scheme, while simultaneously optimizing for the optimal build orientation. Van de Ven et al. [24] applied a method based on multiple overhang filtering steps in order to ensure self-supporting enclosed voids. These methods successfully alleviate the issue of support structures stuck in enclosed voids, but are only suitable for AM methods where no leftover powder or liquid remains in enclosed voids, e.g. for fused filament fabrication.

In this paper a new geometric method to eliminate enclosed voids is proposed for a density-based TO setting, which is less restrictive on the geometry than current geometric solutions. It is defined as a filter based on a flood fill algorithm combined with the cumulative sum approach used previously in a multi-axis machining filter by Langelaar [6]. It uses the current density field as input and produces a density field where all unreachable void elements are converted to solid. The filter is a purely geometrical operation, for which no auxiliary PDEs are solved. It therefore requires very little additional computational effort to determine the new density field. It is possible to achieve direct control over powder removal features through a combination with the morphology operators proposed by Sigmund [25], as these can be used to impose a minimum feature size on void regions. The operation of the filter is semi-differentiable, but this is shown to not have a significant impact on the final topology or convergence. The method is illustrated here using structured meshes, but the formulation is general and can be applied to an unstructured mesh as well.

While a flood fill algorithm has not been used in topology optimization before, it is well established in drawing programs [26–28]. Its most famous use is the bucket fill tool, where it works through a queue of pixels with an initial colour, and changes them to a different specified colour. Anytime a pixel is processed from the queue its neighbours with the same initial colour are added to it. This allows any connected area of pixels with the same colour to be changed to a different colour.

This paper is organized as follows: Firstly in Section 2, the flood fill algorithm is presented, including sensitivity analysis and implementation aspects. In Section 3 the optimization formulations, problems and settings are introduced, after which the proposed enclosed void prevention filter and the effect of its parameters is studied in Section 4. The paper ends with a discussion in Section 5, followed by conclusions in Section 6.

2. Cumulative sum flood fill algorithm

2.1. Working principle

The purpose of the proposed filter is to change any enclosed void to solid, thus forcing the optimizer to either change the shape to be without enclosed voids, or to generate a path towards the enclosed void if it is deemed too important to remove. This is achieved by utilizing a cumulative sum flood fill algorithm shown in Algorithm 1 that loops over all elements.

input : ρ *Unflooded density field*
output: ξ *Cumulatively summed density field*

initialize $\mathcal{Q}$ with boundary elements ; // $\mathcal{Q}$ = `Queue`
$\xi_i = \rho_i$ for i in $\mathcal{Q}$
$\mathcal{P} = \mathcal{Q}$; // $\mathcal{P}$ = `Processed elements`
while $\mathcal{Q} \neq \{\}$ **do**
  Find element i = argmin($\{\xi_i \quad \forall\, i \in \mathcal{Q}\}$)
  $\mathcal{A}$ = elements adjacent to i and not in $\mathcal{P}$ **for** j *in* $\mathcal{A}$ **do**
    $\xi_j = \rho_j + \xi_i$
    add j to $\mathcal{Q}$ and $\mathcal{P}$
  end
  remove i from $\mathcal{Q}$
end

Algorithm 1: Pseudo-code of the cumulative sum flood fill method

The algorithm works by considering a queue of elements $\mathcal{Q}$, from which it always processes the lowest density element first. The queue is initialized with a prescribed set of elements, which in this paper are all elements reachable from outside the design space. Whenever an element is processed all adjacent unprocessed elements $\mathcal{A}$ are added to the queue, with their density increased through summation with the density of the element currently being processed. Adjacent elements

1	1	1	1	1	0.1	1
1	0	1	1	0.1	0.1	1
1	1	1	1	1	1	1

(a) Input design

15	16	17	18	19	20	21
8	9	10	11	12	13	14
1	2	3	4	5	6	7

(b) Element numbering

Fig. 2. Initial conditions used in the visualization of the flood fill method.

are defined as sharing an edge in 2D or a face in 3D. The list $\mathcal{P}$ tracks which elements have already been in the queue, so that any element in $\mathcal{P}$ cannot be added another time. The result of this is that any void element requires a path of void elements towards it from the prescribed boundaries to remain void in the flooded density field. If this path does not exist the density of the void element is increased by the amount of solid or intermediate density elements that were processed on the way. For the given input design and element numbering shown in Figs. 2, 3 shows several steps of the cumulative sum portion of this algorithm, for a sample with both an accessible and an enclosed void.

This algorithm does result in a density field with values above one, which is fixed by performing a smooth minimum projection on the cumulatively summed density field:

$$\phi_i = (\xi_i^{-q} + 1)^{-\frac{1}{q}} \quad \forall\, i \in \{1, \dots, N\}\,. \tag{1}$$

Here ϕ_i denotes a flooded element density and N is the total amount of elements. The steepness of the smooth minimum function can be scaled with the parameter q, as shown in its plot given in Fig. 4. It shows that for $q = 1$ we expect intermediate densities, but also retain sensitivity information for many elements into the cumulative sum. Higher values of q will trade this sensitivity information for fewer intermediate densities.

Applying the smooth minimum function finally achieves the desired flooded density field. Fig. 5 shows how the two main steps of this filter influence a given input density field.

2.2. Sensitivity analysis

Sensitivities for this method can be incorporated into any existing gradient based optimization formulation with objective f by applying the chain rule:

$$\frac{\partial f}{\partial \boldsymbol{\rho}} = \frac{\partial f}{\partial \boldsymbol{\phi}} \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{\rho}}. \tag{2}$$

Here the term $\frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{\rho}}$ needs to be determined for this method, which can be done by investigating the steps in Algorithm 1. Using the chain rule once again, the sensitivities to be determined can be separated into the sensitivities for the cumulative sum and the smooth minimum portions of the algorithm.

$$\frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{\rho}} = \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\rho}}. \tag{3}$$

The value of the cumulatively summed densities is determined whenever an element is added to the queue. This value then only depends on the element currently being processed and its own density. However, since the value of the element currently being processed also

1	1	1	1	1	0.1	1
1	0	1	1	0.1	0.1	1
1	1	1	1	1	1	1

(a) Outer boundary added to queue $\mathcal{Q}$.

1	1	1	1	1	**0.1**	1
1	0	1	1	0.1	**0.2**	1
1	1	1	1	1	1	1

(b) Result of the first flood fill step, where $\{i, j\} = \{20, 13\}$.

1	1	1	1	1	0.1	1
1	0	1	1	**0.3**	**0.2**	1
1	1	1	1	1	1	1

(c) Result of the second flood fill step, where $\{i, j\} = \{13, 12\}$.

1	1	1	1	1	0.1	1
1	0	1	**1.3**	**0.3**	0.2	1
1	1	1	1	1	1	1

(d) Result of the third flood fill step, where $\{i, j\} = \{12, 11\}$.

1	1	1	1	1	0.1	1
1	0	1	1.3	0.3	0.2	1
1	1	1	1	1	1	1

(e) Result of the fourth flood fill step, where $\{i\} = \{1\}$, which did not have any adjacent untouched elements. For this step all lowest density elements in the queue had a density value of 1, so this and the following steps were chosen based on lowest element number.

1	1	1	1	1	0.1	1
1	**1**	1	1.3	0.3	0.2	1
1	**1**	1	1	1	1	1

(f) Result of the fifth flood fill step, where $\{i, j\} = \{2, 9\}$.

1	1	1	1	1	0.1	1
1	1	**2**	1.3	0.3	0.2	1
1	1	**1**	1	1	1	1

(g) Result of the sixth flood fill step, where $\{i, j\} = \{3, 10\}$.

Fig. 3. Visual representation of the flood fill method, where each element is filled according to their status in the floodfill algorithm. Here a blue fill denotes untouched elements, a yellow fill denotes already modified elements still in the processing queue $\mathcal{Q}$, and a green fill denotes fully processed elements no longer in the processing queue $\mathcal{Q}$. Bold numbers indicate elements that changed status.

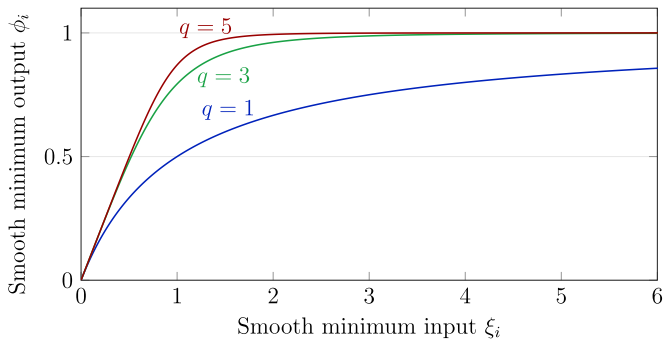


Fig. 4. Smooth minimum projection for different values of q.

1	1	1	1	1	1	1	1	0	0
0	0	1	1	0	0	1	1	1	0
0	0	0	0	1	1	0	0	1	1
0	0	1	1	0	0	1	1	1	0
1	1	1	1	1	1	1	1	0	0

(a) Input density field $\boldsymbol{\rho}$.

1	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	2	2	1	0
0	0	0	0	1	2	2	2	2	1
0	0	1	1	1	1	2	2	1	0
1	1	1	1	1	1	1	1	0	0

(b) Density field after cumulative sum flood fill algorithm $\boldsymbol{\xi}$.

1	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	0
0	0	0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	0	0

(c) Density field after smooth minimum projection $\boldsymbol{\phi}$.

Fig. 5. Visual representation of the two main steps in the flood fill filter, where grey and white denote solid and void elements, and blue denotes void elements of the original design that have been changed to solid due to the filter.

depends on all the elements that were processed leading up to it, these sensitivities also have to carry over to the newly added element. The sensitivities of any new element then become a list of ones and zeroes, where the ones correspond with all elements that were processed leading up to it. Using the same notation as in Algorithm 1 and with k denoting any element the cumulative sum sensitivities become:

$$\frac{\partial \xi_j}{\partial \rho_k} = \begin{cases} 1 & \text{if } k = j \\ \dfrac{\partial \xi_i}{\partial \rho_k} & \text{otherwise.} \end{cases} \tag{4}$$

The fact that all sensitivities of the previously processed elements are also included does mean that $\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\rho}}$ becomes a large sparse matrix. Note that between iterations the order in which elements are processed can change, as it depends on the densities of all elements. These changes in order are not taken into account when calculating the sensitivities, which means that the method is semi-differentiable. The consequence of this is investigated in Section 5.2.

The smooth minimum projection is an element wise operation so the sensitivities can be determined similarly using regular differentiation methods. This gives:

$$\frac{\partial \phi_i}{\partial \xi_i} = \xi_i^{-q-1} \left(\xi_i^{-q} + 1 \right)^{-\frac{1}{q}-1}. \tag{5}$$

2.3. Control over access channel properties

For any void region to exist after the filter an access channel is required of at least one element wide. Depending on the mesh size one element could be too small to be able to remove trapped powder or liquid, in that case control over the minimum size of the void area is necessary. Morphology operators by Sigmund [25] can be used for this purpose by first performing a dilation, shown in Eq. (6), of the solid domain before the floodfill. This step ensures any void smaller than the specified morphology operator radius is changed to solid. Afterwards the erosion step, shown in Eq. (7), is performed on the flooded density field.

$$\tilde{\rho}_d = \ln\left(\frac{\sum_{i \in N_e} e^{\beta \rho_i}}{\sum_{i \in N_e} 1} \right) / \beta. \tag{6}$$

$$\tilde{\rho}_e = 1 - \ln\left(\frac{\sum_{i \in N_e} e^{\beta (1-\rho_i)}}{\sum_{i \in N_e} 1} \right) / \beta. \tag{7}$$

These two equations are a smooth maximum and minimum operator within a certain prescribed neighbourhood N_e. The β parameter controls the steepness of this smooth approximation. Continuation on β is performed to improve convergence and results in this paper will follow a similar continuation strategy as described by Sigmund [25], with an initial β value of 0.2, which gradually increases until a value of 200. In this work the value is increased by 3% every iteration, as opposed to doubling it every 50 iterations.

3. Problem definition

3.1. Optimization formulation

The flood fill algorithm is defined as a filter, which presents multiple ways to implement the method. In this paper three different options will be considered: only using it as a filter, only using flooded densities in the volume constraint and applying a geometric constraint to limit the difference between the flooded and regular densities. These implementations will be applied to well-known compliance minimization topology optimization problems.

Throughout the optimization simplified isotropic material interpolation with penalization (SIMP) is used to suppress the use of intermediate densities. In this paper the lower bound to avoid a singular stiffness matrix is incorporated within the SIMP formulation, as proposed by Sigmund [25]. Eq. (8) shows the SIMP implementation to obtain element Young's modulus E_e, where the input value s can denote either the flooded or unflooded element density, depending on what implementation method is chosen. E is the material Young's modulus and p is the penalization parameter.

$$\begin{aligned} &E_e(s) = E_{\min} + \left(E - E_{\min}\right) s^p \\ &\text{with} \qquad 0 \le s \le 1. \end{aligned} \tag{8}$$

The density filtering method as proposed by Bruns and Tortorelli [29] is also used in this paper, which maps the design variables $\mathbf{x}$ to the filtered design variables $\boldsymbol{\rho}$.

When the algorithm is used purely as a filter then the flooded densities in each element e are used for determining both the stiffness matrix $\mathbf{K}$ and the volume V with its maximum allowed value of $V_{\max}$.

The stiffness matrix is in turn used to determine the displacements $\mathbf{u}$ from the applied loads $\mathbf{f}$.

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{f}^{\mathrm{T}}\mathbf{u} \\ \text{s.t.} \quad & \mathbf{K}(\boldsymbol{\phi}(\boldsymbol{\rho}))\,\mathbf{u} = \mathbf{f} \\ & \frac{V(\boldsymbol{\phi}(\boldsymbol{\rho}))}{V_{\max}} - 1 \leq 0 \\ & 0 \leq x_e \leq 1. \end{aligned} \tag{P1}$$

When the flooded densities are used for the stiffness matrix any filled up enclosed void still adds stiffness to the overall structure, therefore the second approach, shown in Eq. (P2) is only using flooded densities for the volume constraint. This causes any enclosed void filled up by the flood fill to only add unnecessary mass to the result, while offering nothing in terms of stiffness. The intermediate densities in the flood filled density field caused by lower values of q can cause the true volume V_t of the unflooded density field to exceed the required maximum. For instance, when the cumulatively summed density (ξ_e) of an element is equal to 1, then its flooded counterpart (ϕ_e) will be equal to 0.5 for $q = 1$, as shown in Fig. 4. Therefore, the true volume value will be reported for any results using this problem formulation. This problem formulation also only works when adding more mass always results in a better objective, which is the case with compliance minimization.

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{f}^{\mathrm{T}}\mathbf{u} \\ \text{s.t.} \quad & \mathbf{K}(\boldsymbol{\rho})\,\mathbf{u} = \mathbf{f} \\ & \frac{V(\boldsymbol{\phi}(\boldsymbol{\rho}))}{V_{\max}} - 1 \leq 0 \\ & 0 \leq x_e \leq 1. \end{aligned} \tag{P2}$$

Finally, in the geometric constraint formulation a constraint between the flooded and regular densities is added, which works by imposing a maximum total amount of elements that the regular and flooded density field are allowed to differ. Since the density difference between a flooded and unflooded element can take any sign due the smooth minimum, the difference for each individual element is squared. This function can then be used to describe the full geometric constraint:

$$\sum_{e=1}^{N} \frac{(\phi_e - \rho_e)^2}{N} - \chi \leq 0. \tag{9}$$

Here χ denotes the chosen constraint value and N the total amount of elements. The optimization problem reads:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{f}^{\mathrm{T}}\mathbf{u} \\ \text{s.t.} \quad & \mathbf{K}(\boldsymbol{\rho})\,\mathbf{u} = \mathbf{f} \\ & \frac{V(\boldsymbol{\rho})}{V_{\max}} - 1 \leq 0 \\ & \textstyle\sum_{e=1}^{N} \frac{(\phi_e - \rho_e)^2}{N} - \chi \leq 0 \\ & 0 \leq x_e \leq 1. \end{aligned} \tag{P3}$$

3.2. Optimization problem description

The first problem used to study the flood fill method is a 2D cantilever case as shown in Fig. 6, which is discretized in a mesh of 150 × 50 elements. In general applying an enclosed void constraint on a 2D case is very restrictive on the geometry, as there are no pathways to make towards a void without severely impacting the stiffness. This makes this 2D case an academic stress test for the algorithm, with the added advantage that results are easier to visualize than 3D designs.

The real application for enclosed void prevention for AM concerns 3D structures. Fig. 7(a) shows the problem description of a 3D torsion beam, which is discretized in a mesh of 60 × 20 × 20 elements. The non-design domain is two elements thick and the torsion load is applied as shown in Fig. 7(b). This specific case and discretization is commonly used in the literature to study enclosed void prevention methods [10,16,18,19]. In the torsion beam case the optimizer pushes all material to the edge of the design space to withstand the torsion load. This in turn causes a single large enclosed void in the centre, therefore this problem serves as a good benchmark for the enclosed void prevention algorithm.

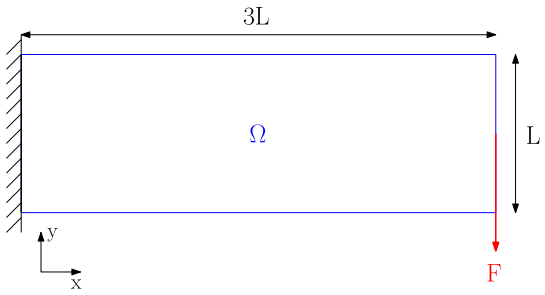
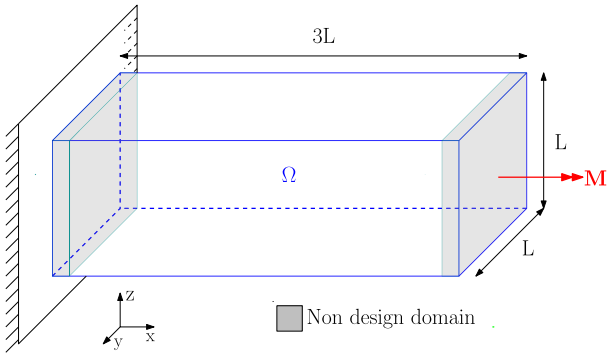
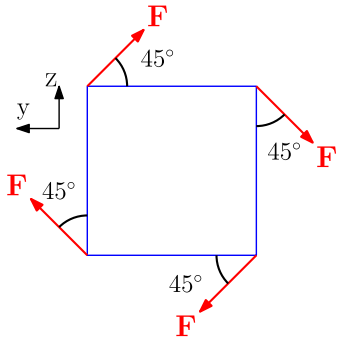


Fig. 6. Problem description of the cantilever beam.



(a) 3D view.



(b) Side view.

Fig. 7. Problem description of the torsion beam.

3.3. Optimization settings

The optimization algorithm used in this work is the method of moving asymptotes by Svanberg [30], with a move limit of 0.1 and a convergence criteria based on the average design variable change between iterations:

$$\frac{1}{N}\sum_{e=1}^{N}\left(\left|x_e - x_e^{\mathrm{old}}\right|\right) \leq \tau. \tag{10}$$

Throughout the optimization specific parameters are used, given in Table 1. Each optimization is performed using a uniform initial density distribution equal to the specified volume constraint.

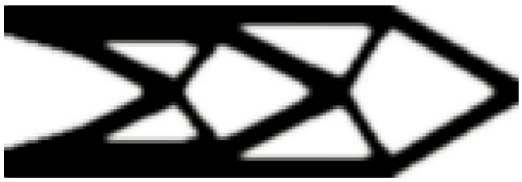


Fig. 8. Free form optimized result for cantilever problem with a reference compliance of $\frac{C}{C_{\text{ref}}} = 100\%$, $V_t = 50\%$ and in 167 iterations.

Table 1
Optimization settings used in numerical examples.

Symbol	Description	Value
r	Radius of density filter	2 elements
p	SIMP penalization parameter	3
E	Ti-4V-6Al Young's modulus	113.9 GPa
ν	Ti-4V-6Al Poisson's ratio	0.342
E_{min}	lower bound SIMP formulation	113.9 kPa
τ_{2D}	stopping criterion 2D problems	10^{-4}
τ_{3D}	stopping criterion 3D problems	$5 \cdot 10^{-4}$
V_{max}	Volume constraint	50%

4. Numerical examples

4.1. 2D cantilever beam

4.1.1. Comparison of problem formulations

First we consider the 2D cantilever problem, for which the free-form solution is given in Fig. 8. This case allows for a visual comparison of the different problem formulations and parameter settings of the flood fill algorithm. The density field used for visualization in all results is what density field is used to determine **K**. In the filter problem formulation results this is the flooded density field ($\boldsymbol{\phi}$), in all other formulations this is the unflooded density field ($\boldsymbol{\rho}$).

The results when using the filter implementation (P1) are given in Fig. 9 for different values of q used in the smooth minimum projection. These results show how the optimization process uses a large internal area of intermediate densities in order to place more material near the top and bottom of the design space, to increase the moment of inertia. In the filter implementation these intermediate densities still offer some stiffness to the overall structure. This solution can therefore be seen as equivalent to an I-beam, where two stiff flanges are connected by a thin web. Unfortunately, these intermediate densities are often not desired from a manufacturing standpoint, which leads to the conclusion that the filter implementation by itself is not a sufficient solution to prevent enclosed voids.

The volume constraint problem formulation (P2) can remedy the issues observed for the filter formulation. The large area of intermediate densities in the previous result would add limited stiffness to the structure, whilst costing a significant amount of material through the flood-filled volume constraint. To increase the efficiency of material usage, the optimization process would have to add material in the unflooded density field. Results for this problem formulation are given in Fig. 10 for different values of q, showing that the optimization process converges to a sideways arch solution which could be considered optimal. This design has also been found in studies on milling constraints in topology optimization [6,7] and previous work on enclosed voids in topology optimization [18]. Another observation is that choosing the value of q represents a trade-off between faster convergence and reduced amount of overshoot of the true volume for lower and higher values of q, respectively. Especially in the results for $q = 1$ the amount of intermediate densities in the flooded density field allow the optimization process to use much more material than the volume constraint originally is supposed to allow, which in turn causes the compliance to be significantly reduced. For additional clarity as to

(a) $q = 1$, $\frac{C}{C_{\text{ref}}} = 205.6\%$, 232 iterations.

(b) $q = 2$, $\frac{C}{C_{\text{ref}}} = 142.2\%$, 164 iterations.

(c) $q = 3$, $\frac{C}{C_{\text{ref}}} = 130.8\%$, 173 iterations.

Fig. 9. Cantilever problem optimized using the filter formulation.

(a) $q = 1$, $\frac{C}{C_{\text{ref}}} = 103.5\%$, $V_t = 61\%$, 137 iterations.

(b) $q = 2$, $\frac{C}{C_{\text{ref}}} = 125.2\%$, $V_t = 51\%$, 201 iterations.

(c) $q = 3$, $\frac{C}{C_{\text{ref}}} = 130.4\%$, $V_t = 50\%$, 286 iterations.

Fig. 10. Cantilever problem optimized using the volume constraint formulation.

(a) Unflooded density field (b) Flooded density field

Fig. 11. Unflooded density field used in stiffness matrix and flooded density field used for volume constraint using $q = 1$.

(a) $q = 1$, $\frac{C}{C_{\text{ref}}} = 103.5\%$, $V_t = 61\%$, 137 iterations.

(b) $q = 2$, $\frac{C}{C_{\text{ref}}} = 125.2\%$, $V_t = 51\%$, 201 iterations.

Fig. 12. Cantilever problem optimized using the geometric constraint formulation with $\chi = 10^{-3}$.

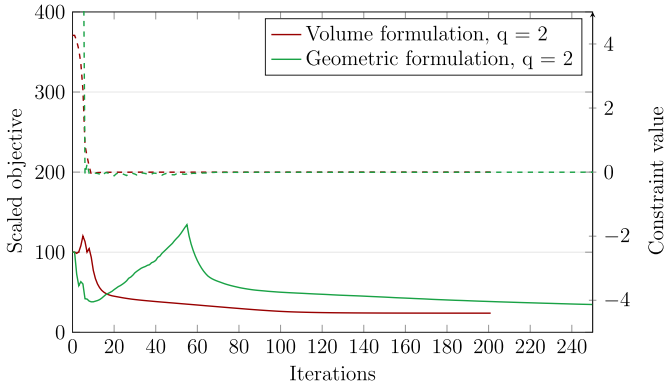


Fig. 13. Convergence plots of representative 2D results, where the solid line denotes the objective and the dashed line denotes the flooded volume constraint or the geometric constraint.

why this occurs, the unflooded and flooded density field are given side by side in Fig. 11.

While the volume constraint problem formulation prevents enclosed voids, it does so by relying on the property that adding material always leads to a better objective. This is a trait of compliance minimization, but is not generally applicable to any topology optimization problem. For alternative problems the geometric constraint problem formulation could be used to prevent enclosed voids. Results using the geometric constraint are given in Fig. 12. The geometric constraint also prevents enclosed voids, but the convergence is negatively impacted by its introduction, as evidenced by the significantly increased number of iterations. This is caused by the geometric constraint severely limiting the possible design changes, which in turn is caused by the reduced amount of sensitivity information available due to the smooth minimum projection.

Iteration histories of the volume and geometric constraint formulation are given in Fig. 13. These show that in the volume constraint formulation the objective initially sharply increases, as the volume constraint is violated due to the large amount of additional material that the flooded density field adds. After this initial sharp increase the objective history shows little to no irregularities. However, this cannot be said for the convergence plot of the geometric constraint, which shows more undesired behaviour. As noted before, this can be explained by the severely limited design changes possible when the geometric constraint is introduced.

Fig. 14. Cantilever problem optimized using the volume constraint formulation with continuation on q, $\frac{C}{C_{\text{ref}}} = 130.5\%$, $V_t = 50\%$, 146 iterations.

(a) Continuation on q, $\frac{C}{C_{\text{ref}}} = 130.5\%$, 511 iterations.

(b) Continuation on χ, $\frac{C}{C_{\text{ref}}} = 133.8\%$, 308 iterations.

(c) Continuation on both q and χ, $\frac{C}{C_{\text{ref}}} = 130.0\%$, 257 iterations.

Fig. 15. Cantilever problem optimized using the geometric constraint formulation with continuation.

4.1.2. Continuation

Continuation can be a viable strategy to benefit from the faster convergence of low values of q while reducing the amount of intermediate densities it causes in later iterations. The continuation strategy used here for the volume constraint problem formulation is to start at a value of $q = 1$ and increase this value by 1% every iteration, until the value of $q = 3$ is reached. The result when applying continuation is given in Fig. 14, which shows that it works well to reduce the amount of iterations until convergence while also reducing intermediate densities.

Convergence for the geometric constraint case in Fig. 12 was very slow due to its limitation on the possible design changes. Using continuation can be very beneficial to remove this limitation, as it will allow larger design changes in early iterations. Continuation could only be applied to q as in the volume constraint formulation, but the geometric constraint formulation allows for continuation on χ as well. For χ the applied strategy is to start at a value of 0.1 and to linearly reduce this by 10^{-3} until it reaches a value of 10^{-3}. This continuation strategy is chosen to allow for some freedom in the initial iterations of the optimization. For q we use the same procedure as described above. Geometric constraint implementation results when applying continuation are given in Fig. 15. These show that the continuation indeed improves convergence speed, where applying it to both the constraint value χ and the projection parameter q shows the largest improvement.

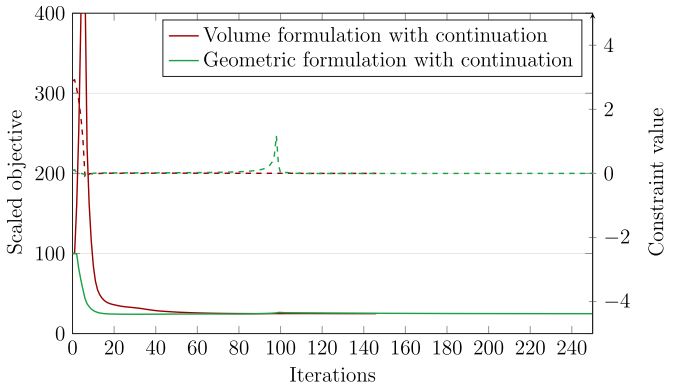


Fig. 16. Convergence plots of 2D results using continuation, where the solid line denotes the objective and the dashed line denotes the flooded volume constraint or the geometric constraint.

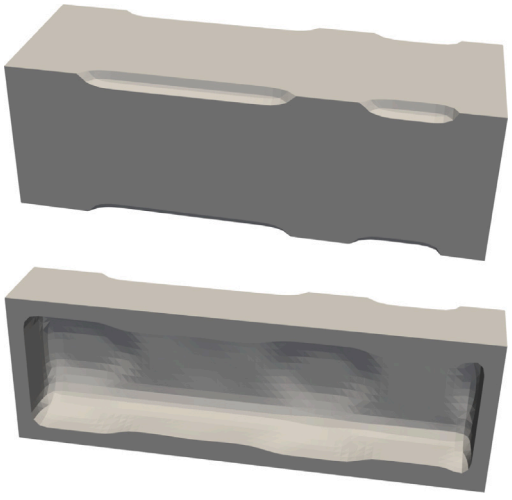


Fig. 17. Free form optimized result for torsion beam problem with a reference compliance of $\frac{C}{C_{\text{ref}}} = 100\%$, $V_f = 50\%$, converged in 110 iterations.

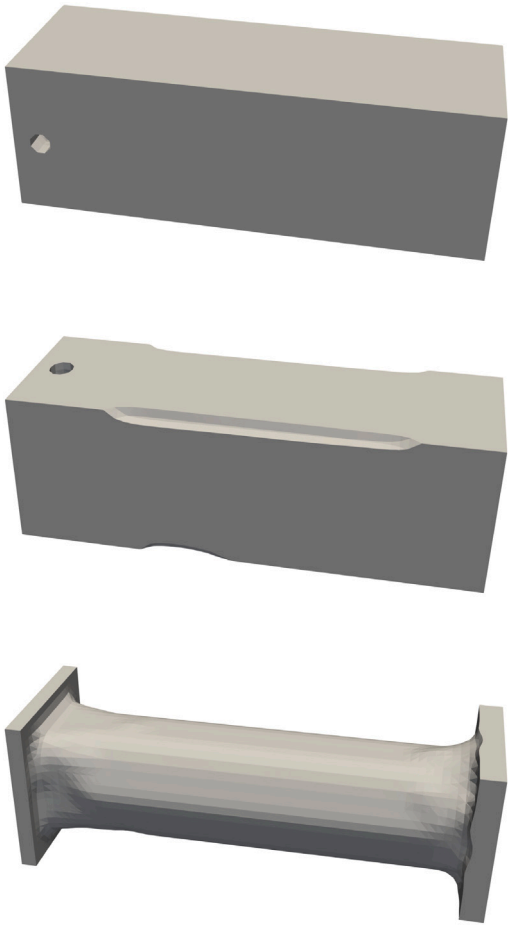


Fig. 18. Optimized results for the torsion beam problem using the volume constraint problem formulation. The interior of designs (a) and (b) are similar to Fig. 17 and a cross-section view is therefore omitted. The interior of (c) is fully solid.

Iteration histories of 2D results using continuation are given in Fig. 16. They show how the same sharp initial increase of the objective for the volume constraint result still occurs with continuation, as our volume constraint is still violated in early iterations. However, the applied continuation on the geometric constraint greatly improved the convergence behaviour.

4.2. 3D torsion beam

4.2.1. Volume constraint formulation

Next we consider the 3D torsion beam problem, which is used to demonstrate that the method can also handle 3D cases. As noted in literature [19] this is easier compared to the 2D cases, as there are more directions in which to create powder removal pathways without drastically impacting the geometry and its performance. This study will therefore focus more on being able to control the access channel properties, for which we use the volume constraint formulation, as it performed the best in the 2D study. The free-form solution to the torsion beam problem is given in Fig. 17. The goal of the enclosed void prevention method for this case is to ensure that the single large enclosed void in the centre of the beam is accessible. All 3D results are displayed with a smoothed density field and with densities below a value of 0.5 removed.

We first present results obtained using only the volume constraint problem formulation (P2). Fig. 18 shows the optimized torsion beam results for different values of q. The resulting designs show that the method successfully makes the enclosed void accessible. When $q = 1$ and $q = 1.5$ are used in the smooth minimum projection the method results in a small access channel towards the large enclosed void in the centre. This single access channel has significantly less impact on compliance when compared to previous work [10,16,18], where the final design had multiple access channels for the single void in the centre. The amount of material used significantly exceeds the volume constraint, which in turn results in a significant reduction in compliance. When $q = 2$ is used, the method pushes all material to the middle of the design space. While this does successfully eliminate enclosed voids, it is also a sub-optimal design. It is suspected that this is related to the reduced amount of sensitivity information available for higher values of q. The initial value for q should therefore not be too large when using this method and all remaining results will use a value of $q = 1$.

Iteration histories of 3D results using the volume constraint formulation are given in Fig. 19. Once again, these show how the objective initially sharply increases due to the violated volume constraint. After this initial sharp increase the objective history again shows no irregularities.

4.2.2. Continuation and additional constraints

The volume constraint results show that it is beneficial for convergence to use $q = 1$, but as observed in the 2D cantilever cases the

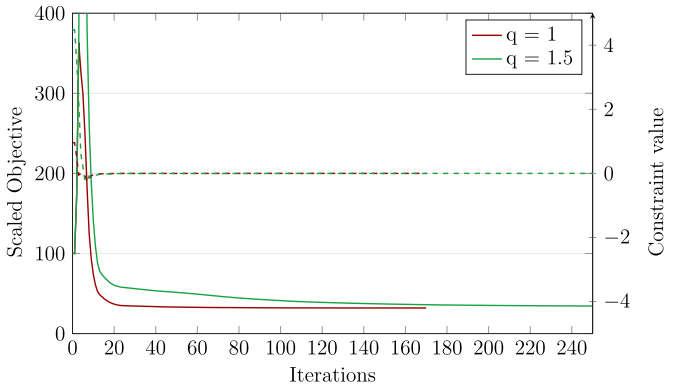


Fig. 19. Convergence plots of 3D results, where the solid line denotes the objective and the dashed line denotes the flooded volume constraint.

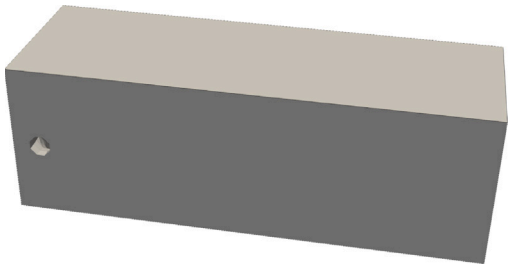


Fig. 20. Torsion beam problem optimized using the volume constraint formulation with continuation on q, $\frac{C}{C_{\text{ref}}} = 98.1\%$, $V_f = 55\%$, 282 iterations. The interior of this design is similar to Fig. 17 and a cross-section view is therefore omitted.

large amount of intermediate densities then allows for more material usage than the volume constraint specifies. To maintain the faster convergence while still only using the specified amount of volume two strategies are considered. The first strategy is continuation on q, in which q is increased by 1% every iteration until a value of 2. Fig. 20 shows the torsion beam result using continuation on q. This result required less iterations to converge to an optimal design than the case with a larger value of q shown in Fig. 18(b). It also reduced the overshoot of the true volume value, but did not entirely eliminate it.

The second strategy is adding an additional volume constraint on the unflooded density field to the volume constraint problem formulation (P2). This additional constraint would ensure that only the specified amount of volume is used:

$$\frac{V(\rho)}{V_{\max}} - 1 \leq 0. \tag{11}$$

The result for this strategy is shown in Fig. 21, which required a number of iterations comparable to only applying a flooded volume constraint with $q = 1$, while not exceeding the total allowed amount of material. This result shows that adding a second volume constraint is an effective strategy to enforce the intended material usage, while still preventing enclosed voids through the floodfill method. All remaining results will therefore use this strategy in order to make them easier to compare with the reference case in Fig. 17 and with each other.

Iteration histories of 3D results using the volume constraint formulation with continuation and additional constraints are given in Fig. 22. They again show similar behaviour to the previous iteration histories using the volume constraint formulation.

4.2.3. Control of amount of access channels

For practical purposes it is beneficial to be able to control the amount of access channels, e.g. so that powder can be removed by

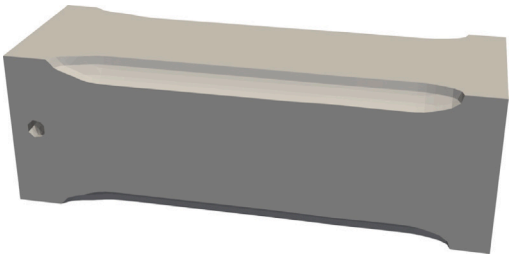


Fig. 21. Torsion beam problem optimized using the volume constraint formulation with an additional volume constraint on ρ, $\frac{C}{C_{\text{ref}}} = 100.9\%$, $V_f = 50\%$, 183 iterations. The interior of this design is similar to Fig. 17 and a cross-section view is therefore omitted.

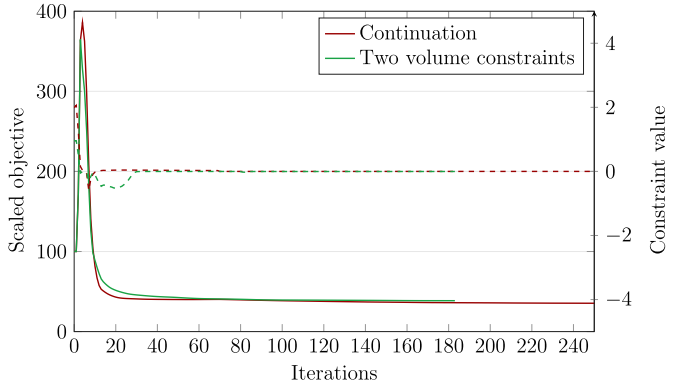


Fig. 22. Convergence plots of 3D results using continuation and an additional constraint, where the solid line denotes the objective and the dashed line denotes the flooded volume constraint.

flowing gas or liquid. The floodfill method can achieve control over the amount of channels through applying multiple independent floodfill steps on the initial unflooded density field. Each of these steps then has their own associated initial boundary elements and volume constraint. For example, if we were to enforce a minimum of two access channels on a design, then we employ three total volume constraints: one on the unflooded density field, one on the first flooded density field with initial boundary A, and finally one on the second flooded density field with initial boundary B. It is important to note that some care is necessary in picking the initial boundary, as allowing them to overlap would allow the optimization process to satisfy both flooded volume constraints using only one access channel.

In the result given in Fig. 23 we enforce a minimum of four access channels. Each of the large rectangular domain boundaries was used as initial flood fill boundary, excluding the elements at the edges such that no elements were present in multiple initial boundaries. The result shows that the flood fill method can achieve control over the minimum amount of access channels.

4.2.4. Control of access channel size

Depending on the mesh size it is possible that an access channel of one element wide is not enough to expel all powder or liquid. For this reason accurate control over the size of the access channel is necessary, which can be achieved through morphology operators as described in Section 2.3. Results using these are given in Fig. 24 for different radii. The results all contain one singular access channel, but as expected its size increases as the radius of the morphology operators increases.

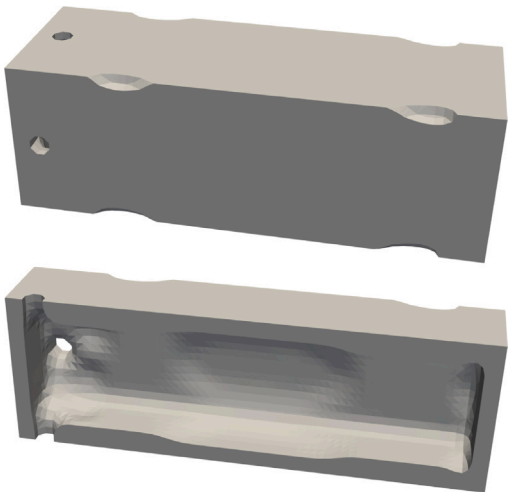
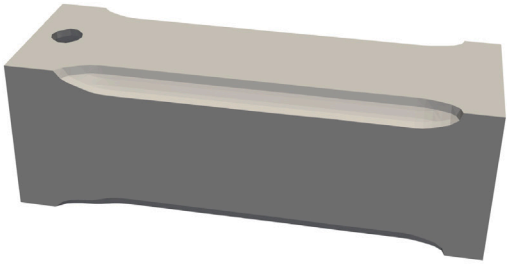
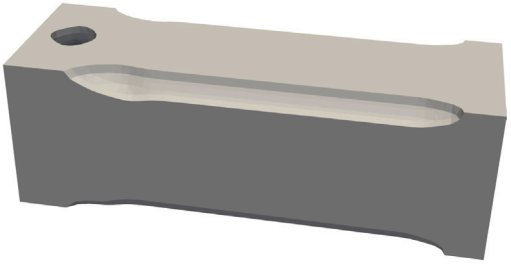


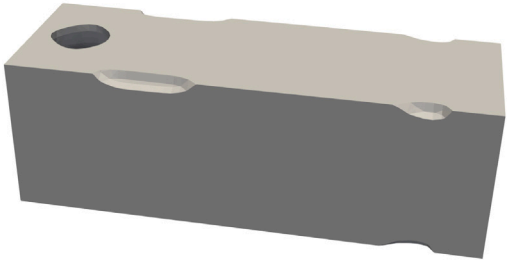
Fig. 23. Optimized results for the torsion beam problem using the modified flood fill algorithm, while enforcing a minimum of four access channels, $\frac{C}{C_{\text{ref}}} = 100.6\%$, $V_t = 50\%$, 107 iterations.



(a) Radius = 1, $\frac{C}{C_{\text{ref}}} = 101.1\%$, $V_t = 50\%$, 184 iterations.



(b) Radius = 2, $\frac{C}{C_{\text{ref}}} = 101.3\%$, $V_t = 50\%$, 153 iterations.



(c) Radius = 3, $\frac{C}{C_{\text{ref}}} = 100.8\%$, $V_t = 50\%$, 114 iterations.

Fig. 24. Torsion beam problem optimized using the volume constraint formulation with morphology operators of different radii to control access channel size. The interior of these designs is similar to Fig. 17 and a cross-section view is therefore omitted.

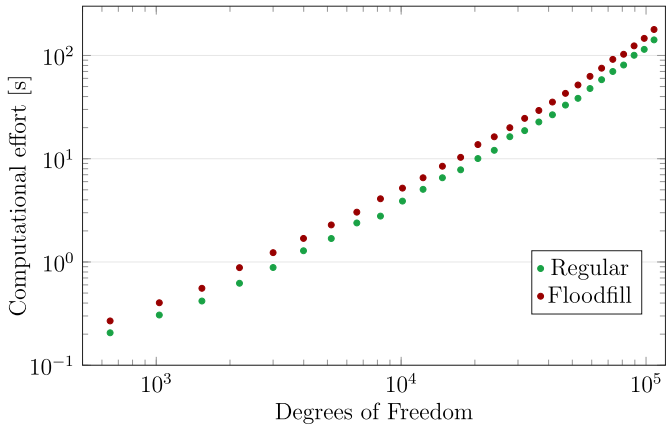


Fig. 25. Measured computational effort of the flood fill algorithm.

5. Discussion

5.1. Computational effort

The computational effort of the method is investigated by taking the average of 5 optimization loops of 10 iterations. These were performed with a laptop using a single core of an Intel Core i7-11850H processor. The algorithm was written in Python using the PyMoto [31] topology optimization framework. A cube of varying size is used as test case, to which two alternatives are applied: Optimization with and without the flood fill. The resulting computational effort is given in Fig. 25, where the dots represent the acquired data. The figure shows that running the optimization with the floodfill algorithm does add some additional computational effort, which scales similarly to the linear solve used in both cases.

It should also be noted that no special measures were taken to optimize the code of the floodfill procedures, and that it was written in an interpreted coding language. Because the method relies on looping over all elements, it will have a significantly reduced computational effort when implemented in a compiled language. The cost of sensitivity analysis can further be reduced by implementing an adjoint sensitivity analysis approach as described in [32].

5.2. Semi-differentiability

As noted in Section 2.2, the order in which elements are processed depends on the element densities of that specific iteration. This part of the algorithm is non-differentiable, which could lead to stability issues during the optimization. We aim to investigate the impact of the semi-differentiability of the algorithm by studying the results of the volume constraint formulations in 2D and 3D. This is done by comparing when any element is added to the queue between iterations. In particular we register whether any element is added to the queue by a different element than in the previous iteration, as this would mean that the dependencies have changed and the sensitivities for that element are no longer correct.

Fig. 26 shows what percentage of elements may have had faulty sensitivity information because of this changed dependency for the 2D results given in Fig. 10. It shows that for different values of q the percentage of elements with faulty sensitivities never surpasses 6%. The plot for the 3D results of Fig. 18 is given in Fig. 27 and it shows a similar error percentage. Furthermore, as the design stabilizes throughout the topology optimization process, the occurrence of these dependency changes quickly diminishes to levels well below 1%. Also considering the regular appearance of the results obtained, it seems that the semi-differentiable nature of the floodfill process has no strong detrimental effect on the optimization process.

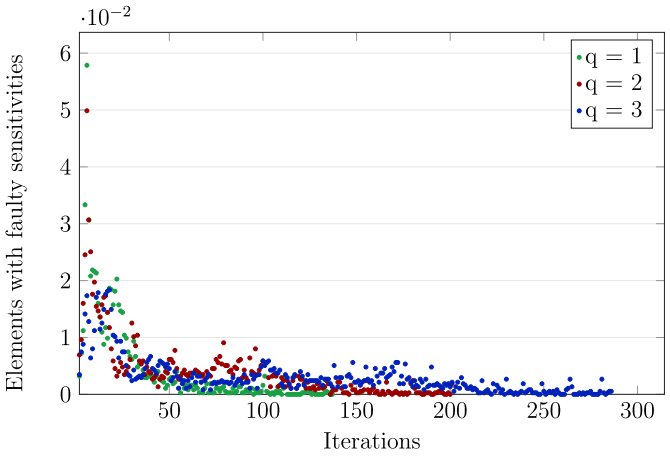


Fig. 26. Fraction of elements with faulty sensitivities in 2D volume constraint results.

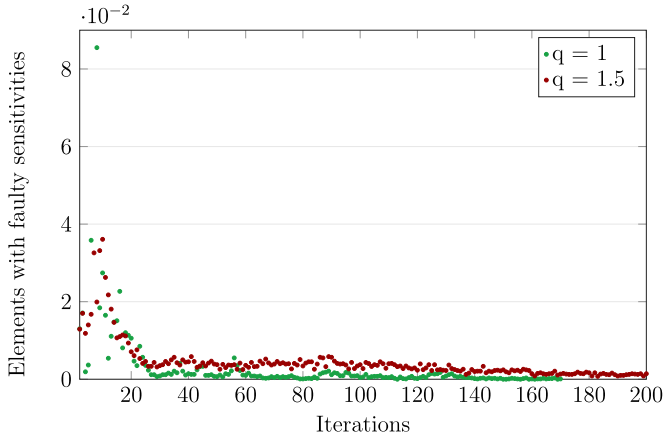


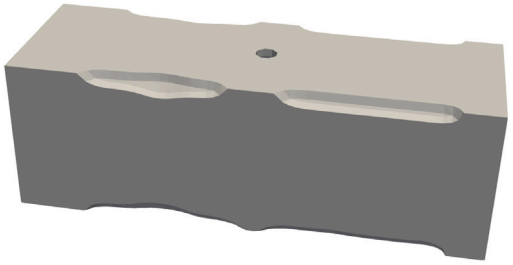
Fig. 27. Fraction of elements with faulty sensitivities in 3D volume constraint results.

5.3. Influence of element numbering

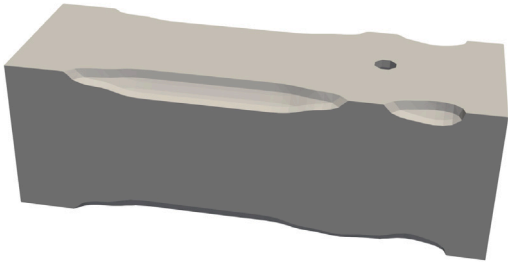
The torsion beam results all had a similar location for their access channel, which begs the question whether or not this is an optimal location for this channel or if this is caused by the initial dependency of the flood fill algorithm on element numbering. We performed two tests to investigate which of these influenced the access channel location. The first test is to apply a predetermined void non-design domain at different locations along the beam, to force the optimizer to utilize that location for its access channel. The results for this test are given in Fig. 28. The compliance of the design in Fig. 21 is slightly higher than the results presented in Fig. 28. However, this was found to be due to the higher number of iterations that resulted in a slightly better 0–1 design. When running the case in Fig. 21 for 230 and 286 iterations, similar to the results in Fig. 28, the design remains the same, however the compliance ratio reduces to 100.7% and 100.4%, respectively.

The second test is to randomize the order in which the boundary elements are added to the initial queue, therefore slightly changing some dependencies. The resulting design when randomizing the order of boundary elements is given in Fig. 29, where the optimizer opts to place the access channel in the same x-location close to the boundary as in the previous results. Different randomizations did result in a different side, i.e. its y, z-coordinates, being chosen for the access channel, due to the problem being symmetric in the $y-z$ plane.

From these tests we can conclude that for this problem, the design is primarily driven by structural reasons instead of any potential bias induced by the floodfill algorithm. The access channels are created near the base as this has minimal impact on structural performance.



(a) Access channel forced to be halfway along the beam, $\frac{C}{C_{\text{ref}}} = 100.4\%$, $V_t = 50\%$, 286 iterations.



(b) Access channel forced to be three-quarters along the beam, $\frac{C}{C_{\text{ref}}} = 100.7\%$, $V_t = 50\%$, 230 iterations.

Fig. 28. Optimized results for the torsion beam problem when enforcing different access channel locations. The interior of these designs is similar to Fig. 17 and a cross-section view is therefore omitted.

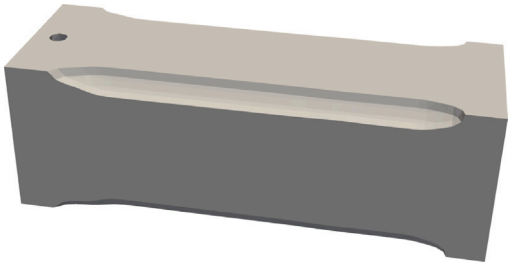


Fig. 29. Torsion beam result when randomizing the order of the boundary elements, $\frac{C}{C_{\text{ref}}} = 100.9\%$, $V_t = 50\%$, 184 iterations. The interior of this design is similar to Fig. 17 and a cross-section view is therefore omitted.

6. Conclusions

In this paper a new filter to eliminate enclosed voids in density-based topology optimization is proposed. It does not rely on additional physical analyses but on a simple flood fill process. The filter fills up any void that cannot be reached from a prescribed boundary, thus forcing a pathway towards these voids when they are deemed important for the objective, or alternatively elimination of voids by filling them with solid material. The filter is applicable in different problem formulations, of which the volume constraint and geometric constraint problem formulations are found to perform better than the direct filter application.

The method involves a smooth minimum projection with a parameter q that can be used to control the steepness of the projection. Choosing the value of this parameter was found to be a trade-off between intermediate densities present in the design and convergence speed. Lower values of q also allow for more material usage than the volume constraint specifies in the volume constraint problem formulation, which can be remedied by an additional volume constraint. Continuation was found to be an effective measure to improve both convergence speed and result definition.

In 3D the method can be used to control the location, amount and size of the void access channels. Their location can be controlled by choosing a different initial described boundary in the algorithm, while their amount can be controlled through running additional flood fills with different prescribed initial boundaries. The size of the access channels can be controlled through any means of enforcing a minimum feature size on the void phase, which was done through morphology operators.

The method has a relatively small impact on the computational effort when compared to the linear solve required for the FEM analysis. The introduction of a semi-differentiable algorithm into gradient based optimization could negatively impact the convergence of the optimization. However, investigation of the number of elements with faulty sensitivities revealed that this affects only a small fraction of the elements, which quickly diminishes during the optimization process.

Currently this method allows for control over the location, size and amount of access channels, but their maximum length or straightness cannot be enforced. Imposing a maximum access channel length could be used to ensure all powder can be removed through it. Another possible extension is to ensure that two access channels to any void are on opposite sides of that void, so that pressurized air or liquid can be efficiently used to expel any leftover powder. The combination of this method with existing additive manufacturing filters or constraints for overhanging features is another important topic for future studies.

Another direction for future research is an alternative use of the flood fill process. Undesirable disconnected solid parts can form in certain topology optimization problems, e.g. in eigenfrequency maximization. These can be detected and eliminated in a similar manner as the enclosed void prevention in the present paper.

CRediT authorship contribution statement

Joran van der Zwet: Conceptualization, Software, Validation, Formal analysis, Investigation, Writing – original draft, Visualization. **Arnoud Delissen:** Conceptualization, Software, Supervision, Writing – review & editing. **Matthijs Langelaar:** Supervision, Writing – review & editing.

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.

Data availability

Data will be made available on request.

References

[1] Xia Q, Shi T, Wang MY, Liu S. A level set based method for the optimization of cast part. Struct Multidiscip Optim 2010;41:735–47.
[2] Lu J, Chen Y. Manufacturable mechanical part design with constrained topology optimization. Proc Inst Mech Eng B 2012;226:1727–35.
[3] Wang Y, Kang Z. Structural shape and topology optimization of cast parts using level set method. Internat J Numer Methods Engrg 2017;111:1252–73.
[4] Liu J, Ma YS. 3D level-set topology optimization: a machining feature-based approach. Struct Multidiscip Optim 2015;52:563–82.
[5] Vatanabe SL, Lippi TN, Lima CR, Paulino GH, Silva EC. Topology optimization with manufacturing constraints: A unified projection-based approach. Adv Eng Softw 2016;100:97–112.
[6] Langelaar M. Topology optimization for multi-axis machining. Comput Methods Appl Mech Engrg 2019;351:226–52.
[7] Lee HY, Zhu M, Guest JK. Topology optimization considering multi-axis machining constraints using projection methods. Comput Methods Appl Mech Engrg 2022;390:114464.
[8] Reddy SN, Maranan V, Simpson TW, Palmer T, Dickman CJ. Application of topology optimization and design for additive manufacturing guidelines on an automotive component. In: Proceedings of the ASME design engineering technical conference, Vol. 2A-2016. American Society of Mechanical Engineers Digital Collection; 2016.
[9] Delissen A, Boots E, Laro D, Kleijnen H, van Keulen F, Langelaar M. Realization and assessment of metal additive manufacturing and topology optimization for high-precision motion systems. Addit Manuf 2022;58:103012.
[10] Liu S, Li Q, Chen W, Tong L, Cheng G. An identification method for enclosed voids restriction in manufacturability design for additive manufacturing structures. Front Mech Eng 2015;10:126–37.
[11] Li Q, Chen W, Liu S, Tong L. Structural topology optimization considering connectivity constraint. Struct Multidiscip Optim 2016;54:971–84.
[12] Luo Y, Sigmund O, Li Q, Liu S. Additive manufacturing oriented topology optimization of structures with self-supported enclosed voids. Comput Methods Appl Mech Engrg 2020;372:113385.
[13] Wang C, Xu B, Meng Q, Rong J, Zhao Y. Numerical performance of Poisson method for restricting enclosed voids in topology optimization. Comput Struct 2020;239:106337.
[14] Yamada T, Noguchi Y. Topology optimization with a closed cavity exclusion constraint for additive manufacturing based on the fictitious physical model approach. Addit Manuf 2022;52:102630.
[15] Sabiston G, Kim IY. Void region restriction for additive manufacturing via a diffusion physics approach. Internat J Numer Methods Engrg 2020;121:4347–73.
[16] Donoso A, Aranda E, Ruiz D. A new approach based on spectral graph theory to avoiding enclosed holes in topology optimization. Comput Methods Appl Mech Engrg 2022;393:114769.
[17] Donoso A, Aranda E, Ruiz D. A continuous model for connectivity constraints in topology optimization. Struct Multidiscip Optim 2023;66:1–9.
[18] Zhou L, Zhang W. Topology optimization method with elimination of enclosed voids. Struct Multidiscip Optim 2019;60:117–36.
[19] Gaynor AT, Johnson TE. Eliminating occluded voids in additive manufacturing design via a projection-based topology optimization scheme. Addit Manuf 2020;33:101149.
[20] Xiong Y, Yao S, Zhao ZL, Xie YM. A new approach to eliminating enclosed voids in topology optimization for additive manufacturing. Addit Manuf 2020;32:101006.
[21] Liu D, Chiu LN, Davies C, Yan W. Topology optimization incorporating a passageway for powder removal in designs for additive manufacturing. Struct Multidiscip Optim 2022;65:1–17.
[22] Luo Y, Sigmund O, Li Q, Liu S. Topology optimization of structures with infill-supported enclosed voids for additive manufacturing. Addit Manuf 2022;55:102795.
[23] Wang C. Simultaneous optimization of build orientation and topology for self-supported enclosed voids in additive manufacturing. Comput Methods Appl Mech Engrg 2022;388:114227.
[24] van de Ven E, Ayas C, Langelaar M, Maas R, van Keulen F. Accessibility of support structures in topology optimization for additive manufacturing. Internat J Numer Methods Engrg 2021;122:2038–56.
[25] Sigmund O. Morphology-based black and white filters for topology optimization. Struct Multidiscip Optim 2007;33:401–24.
[26] Pavlidis T. Filling algorithms for raster graphics. Comput Graph Image Process 1979;10:126–41.
[27] Lee ET, Pan YJ, Chu P. An algorithm for region filling using two-dimensional grammars. Int J Intell Syst 1987;2:255–63.
[28] Burtsev SV, Kuzmin YP. An efficient flood-filling algorithm. Comput Graph 1993;17:549–61.
[29] Bruns TE, Tortorelli DA. Topology optimization of non-linear elastic structures and compliant mechanisms. Comput Methods Appl Mech Engrg 2001;190:3443–59.
[30] Svanberg K. The method of moving asymptotes—a new method for structural optimization. Internat J Numer Methods Engrg 1987;24:359–73.
[31] Delissen A. pyMOTO: Modular framework for topology optimization with semi-automatic derivatives. 2023.
[32] Langelaar M. An additive manufacturing filter for topology optimization of print-ready designs. Struct Multidiscip Optim 2017;55:871–83.