C. R. Mecanique 346 (2018) 1104–1121

Contents lists available at ScienceDirect

Comptes Rendus Mecanique

www.sciencedirect.com

Computational methods in welding and additive manufacturing/Simulation numérique des procédés de soudage et de fabrication additive

Using a virtual skeleton to increase printability of topology optimized design for industry-class applications

Yoram Mass [*], Oded Amir

Faculty of Civil Engineering, Technion – Israel Institute of Technology, Technion City, Haifa 3200003, Israel

ARTICLE INFO

Article history:
Received 6 June 2018
Accepted 21 June 2018
Available online 16 August 2018

Keywords:
Topology Optimization
Additive manufacturing
Overhang angles
3-D printing
3-D design
Virtual skeleton

ABSTRACT

This article broadens the scheme previously developed to associate topology optimization with additive manufacturing through the use of a virtual skeleton, consisting in solving the same physical problem with a discrete approach and then with a continuum one. This procedure for 3D designs is applied to various domain geometries, to demonstrate its pertinence on high-resolution industrial cases. An algorithm searching for the best printing direction, exploring the solid angle, is also described and validated; the surface-shaped presentation of the result allows immediate understanding of the influence of the discrete problem parameters, while its running time is much lower than a unique continuum optimization simulation, which proves the attractiveness of the method. In the three examples studied, the procedure outputs exhibit greater printability than the ones produced by traditional methods in each of the printing direction tested, albeit responsibility is left to the final user to choose his best trade-off. Furthermore, the unprintable zones are readily displayed to be either reworked or supported. Explanations about increase of convergence likelihood on discrete structures despite the geometry complexity of an industrial application are also provided and demonstrated.

1. Introduction

The first additive manufacturing (AM) device was patented in the 1980s [1] and intended for stereolithography. The name has remained in the file extension used – STL, but the breakthrough has evolved and now, a wide range of materials can be used by this technology [2–5], in an extensive variety of application fields [6–10]. Although AM aggregates several methods and techniques, most of the academic works on the topic today concern the 3D printing approach [11]. Research is mainly directed toward finding new materials and their related printing processes and simulations. This modus operandi allows the production of almost free-form structures, hence is indicated to manufacture complex designs, such as the ones outputted by topology optimization (TO) algorithms.

TO aims at apportioning a minimal amount of material an optimal manner in a given design domain. Since the seminal paper from Bendsøe and Kikuchi [12], this field has developed from a load-bearing application to diverse engineering and science fields [7–9], generating various formulations and approaches [6]. Due to the freedom of the output topology, optimal designs are in general too complex to be built by conventional manufacturing processes. Consequently, it seems that AM

[*] Corresponding author.
E-mail addresses: yoram.mass@technion.ac.il (Y. Mass), odedamir@technion.ac.il (O. Amir).

https://doi.org/10.1016/j.crme.2018.08.005

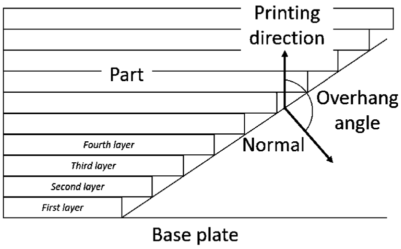


Fig. 1. Location of the overhang angle.

technologies are a perfect match to deal with TO designs. However, one still needs to consider the 3D printing limitations, such as the overhangs.

An overhang is the angle formed between the surface normal of a design and the building direction – or the baseplate modulo π – from which printing is performed with less precision, if possible at all. The location of this angle is illustrated in Fig. 1. Its value varies according to the materials and technologies used. No matter which material is used, although for different physical reasons, hanging elements without supports cannot be produced, or at least with poor precision or surface finish. Those problems result from the layer-fashioned manufacturing as depicted in Fig. 1. The greater the overhang angle, the more pronounced the stair-like shape appear. This issue is known as the *staircase effect*. Along with a *warping effect* due to the heat removal inefficiency in the structure, it constitutes the most challenging reason explaining the need for support addition during manufacturing. Both effects can be minimized by determining the best printing direction; this lead has been explored extensively [13–15], especially for stereolithography. All along this article, particularly for the numerical experiments and without any loss of generality, the maximum overhang angle will be assumed to be 45° – or 135° if taken from the printing direction [16]. Features with greater overhang than that value will be assumed to need support components.

To the authors' best knowledge, the first to deal with overhangs were Brackett et al. [17], who proposed an algorithm to localize them and suggested a way to implement an overhang constraint. Then, Leary et al. [18] introduced the idea of self-supporting topologies, where the supports would be part of the design. The actual implementation consisted in modifying the final design so as to make it printable, modifying at the same time the total amount of material. Subsequently, most of the research publications aimed at obtaining self-supported designs. Several approaches emerged.

Filtering is one of the most popular techniques to integrate supports to a part. The main advantage is that the formulation does not change, then no constraints are added. This approach degrades only slightly the solution optimality. On the other hand, a computational burden, in general small, is added. The works from Guest and Gaynor [19,20] introduce a filter that scans the way each element is supported and penalizes the ones which are not, at each iteration. Langelaar [21,22] then uses the concept to implement a filter that imitates the printing process. Van de Ven et al. [23] developed another filter, which mimics the propagation of a wave and penalizes the zones where the wave arrives late.

Other researchers choose to enforce the overhang limitation explicitly through a change of the formulation or approach. Guo et al. [24] propose a method named "Method of Moving Components" (MMC) together with its companion, the Method of Moving Voids (MMV), that uses predefined geometries whose dimension and location are optimized and projected onto a continuum design domain to form the final structure. Amir and Mass [25] choose to simulate the physical limitation of the fused deposition of plastic by adding an artificial self-weight of the material, at each iteration and for a group of layers. The idea is to force the design to be self-supported. Allaire et al. employ the same idea with the level-set approach [26], along with a second formulation that considers a direct constraint on the angle. Finally, the authors recently suggested a method laying on the use of a virtual truss skeleton to privilege specific locations known as efficient [27].

From another point of view, some studies target the supports themselves. Mirzendehdel and Suresh [28] propose to optimize simultaneously the topology and the supports, without any direct claim regarding the overhangs. The support volume is constrained, but while forced to be zero on the examples given, all the holes are removed from the topology. A recent publication from Qian [29] uses edge detectors to control the undercut and the minimal overhang angle without motivation regarding compliance.

The work presented in this article extends the technique published in [27], where focus was on the underlying formulation and 2D implementation. The important increase in the number of variables implies the use of appropriate tools and the correct implementation of the concept. The optimization problem is defined by a discrete formulation: a coarse grid of nodes is constituted along with bars connecting them – this is called the *ground structure*, and the bar cross-sections are optimized to solve the problem. Then, the truss obtained is rastered and exported into a design space constituted by a regular grid of finite elements – this is the *fine grid*, where its location will be given artificial extra stiffness to privilege material deposition. It is worth noting that, at this stage, the truss is nothing more than an educated guess and "a piece of advice" for continuum optimization, depending on an influence parameter α. In this publication, we aim at proving that the high-resolution 3D implementation of the method is appropriate for industrial applications. In particular, one of the proposed extensions is a search for the best printing direction algorithm. Several printing directions are tested in the solid angle and the best one is selected to serve as a guide for the continuum optimization.

The remainder of the article is organized as follows. The problem formulation is stated in Section 2, as well as the main formulas and relations to constitute the sensitivities needed for a gradient-based optimization. Then, the methodology and its application are explained and detailed through implementation, after having mentioned the main evolution in the tooling aspect. Numerical examples are performed in Section 3, where three situations are explored: for a convex, a non-convex and a more general design domain, inspired by realistic industrial cases. Finally, conclusions are discussed in Section 4.

2. Problem statement

2.1. Formulation

The aim of this article is to extend both the method published in [27] and its functions to general 3D designs. The approach is a two-step method, applied on a unique topology optimization problem. In the first step, a discrete formulation of the problem is solved by a truss, an assemblage of straight bars connected by hinges at their extremities, in which only axial force can develop. A ground structure is created and the cross-sections of its elements are optimized by solving problem (1). In this formulation, we want to find the stiffest structure by minimizing the strain energy $f = \mathbf{u}^\mathsf{T}\mathbf{K}\mathbf{u}$ within the elastic assumption $\mathbf{f} = \mathbf{K}\mathbf{u}$. This latter energy is equal to the *compliance* $\mathbf{f}^\mathsf{T}\mathbf{u}$, which will be used as a structural performance measure all along this article. Furthermore, in all the formulations, f is the objective function and g_j are the j constraints of the problem.

$$\begin{cases} \min\limits_{\boldsymbol{\phi}} f = \mathbf{f}^\mathsf{T}\mathbf{u}_1 \\ \text{s.t.} \begin{cases} g = \sum\limits_{i=1}^{nbars} \phi_i l_i a_i - V^* \leq 0 \\ \phi_i \geq 0 \quad i = 1, \ldots, nbars \end{cases} \\ \text{with:} \quad \mathbf{K}\mathbf{u}_1 = \mathbf{f} \end{cases} \tag{1}$$

where $\boldsymbol{\phi}$ is the vector of the scaling factors of the bars' cross-section areas, one for each truss member; $\mathbf{f}$ is the applied force; $\mathbf{u}_1$ is the nodal displacements vector; l_i and a_i are the length and the cross-section area of the i-th bar, respectively; V^* is the maximal allowed total volume; and $\mathbf{K}$ is the stiffness matrix, computed as $\mathbf{K} = \sum_{i=1}^{nbars} \phi_i \frac{a_i E_i}{l_i} \mathbf{k}_i$ where $\mathbf{k}_i$ encloses only geometric properties of the i-th bar and does not depend on the design variable, and E_i is the Young's modulus of this bar. This formulation is all the more interesting that it clearly shows that the same problem is solved twice – for discrete and continuum ground structures. Nevertheless, this problem is also non-linear, and it has been demonstrated [6,30] that it was equivalent to the following linear one:

$$\begin{cases} \min\limits_{\mathbf{a},\mathbf{q}} f = \sum\limits_{i=1}^{nbars} (a_i l_i) \\ \text{s.t.} \begin{cases} \mathbf{B}\mathbf{q} = \mathbf{f} \\ a_i \sigma^{\min} \leq q_i \leq a_i \sigma^{\max} \quad i = 1, \ldots, nbars \\ a_i \geq 0 \quad i = 1, \ldots, nbars \end{cases} \end{cases} \tag{2}$$

with a_i the area of bar number i, components of vector $\mathbf{a}$; l_i length of the i-th bar; $\sigma^{\min}$ and $\sigma^{\max}$, respectively, the minimum and maximum admissible stress in the bars; q_i the internal force of bar number i, constituting vector $\mathbf{q}$ of internal member forces, in the directions of the truss members; $\mathbf{f}$ is the vector of external forces at the nodes, along the directions of the global degrees of freedom; $\mathbf{B}$ is the compatibility matrix that defines the force equilibrium. Essentially, $\mathbf{B}$ contains geometric information regarding the bars' directions. As the problem is linear, it can be solved by Linear Programing techniques, which are used in this research and the solution is an optimal discrete structure: a truss.

This optimal truss is then rasterized into a vector $\mathbf{T}$ and exported into the discretized continuum domain, where it will influence the material repartition by privileging specific locations. This preference is realized by imposing a greater Young's modulus at points of interest, where T_i, the components of $\mathbf{T}$, will be 1, indicating the material in the truss. In the second step, the optimization problem is expressed by a continuum formulation in a density-based approach.

A robust formulation based on the Heaviside projections [31,32] has been chosen for the implementation. In this method, the initial densities are first filtered by a density filter [33,34] given by (3):

$$\widetilde{\rho}_i = \frac{\sum\limits_{j \in N_i} w(\mathbf{x}_j) v_j \rho_j}{\sum\limits_{j \in N_i} w(\mathbf{x}_j) v_j} \tag{3}$$

with a linear weighting function

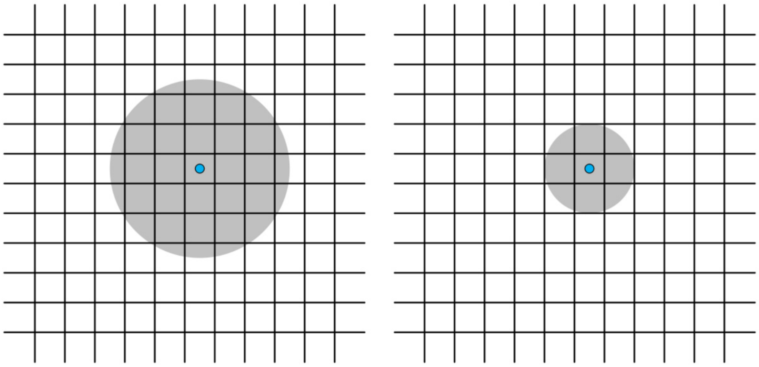


Fig. 2. Affected cells by a filter of radiuses $r_{\min} = 3$ (left) and $r_{\min} = 1.5$ (right).

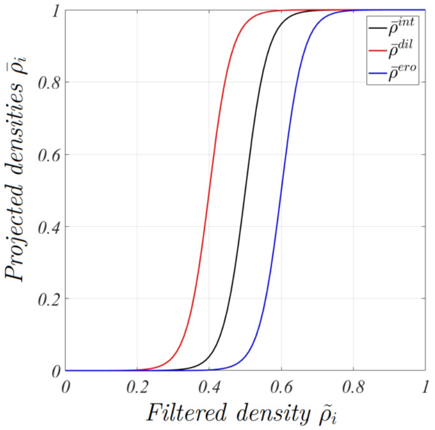


Fig. 3. Heaviside filter effect.

$$w(\mathbf{x}_j) = r_{\min} - \left\|\mathbf{x}_j - \mathbf{x}_i\right\| \tag{4}$$

where $r_{\min}$ is the specified filter radius, $\mathbf{x}_i$ is the position of the centroid of element i and $\mathbf{x}_j$ is the position of the centroid of element j, which is in the neighborhood N_i of element i and thus is assigned a positive weight $w(\mathbf{x}_j)$. Figure 2 illustrates how the elements are affected in a regular mesh as a function of the radius. This distribution is then effectively projected and the behavior of the element density $\bar{\rho}_i$ – the filter output – is ruled by the parameters η, the projection threshold, and β, the curvature of the smooth projection. The effect of this projection is depicted in Fig. 3 and expressed in Eq. (5).

$$\bar{\rho}_i = \frac{\tanh(\beta\eta) + \tanh(\beta(\tilde{\rho}_i - \eta))}{\tanh(\beta\eta) + \tanh(\beta(1-\eta))} \tag{5}$$

where $\tilde{\rho}_i$ is the i-th element density of the design after density filter operation. If $\eta = 0.5$, the function is symmetric in the range of ρ: intermediate densities between 0 and 0.5 will be “pushed” toward 0 and the ones between 0.5 and 1 will be “pushed” toward 1. If $\eta > 0.5$, more intermediate densities will be pushed toward 0, then the filter will produce eroded designs, and conversely. In the specific robust form, control over the minimum length scale is enforced by computing worst-case designs through eroded and dilated ones [31]. Compliance calculation is based on the eroded design, as the latter would maximize it, while the constraint and stiffness computations are based on the dilated design [32].

The final formulation of the continuum optimization problem is written in (6),

$$\begin{cases} \min\limits_{\boldsymbol{\rho}} f(\bar{\boldsymbol{\rho}}, \eta_{\text{ero}}) = \mathbf{f}^{\mathsf{T}}\mathbf{u}_2 \\ \text{s.t.} \begin{cases} g(\bar{\boldsymbol{\rho}}, \eta_{\text{dil}}) = \sum\limits_{i=1}^{nelem} (\bar{\rho}_i v_i) - V^{**} \leq 0 \\ 0 \leq \rho_i \leq 1, \quad i = 1, \ldots, nelem \end{cases} \\ \text{with:} \quad \sum\limits_{i=1}^{nelem} \left(E_{\text{eff},i}(\bar{\rho}_i, \eta_{\text{ero}})\mathbf{k}_i^0\right)\mathbf{u}_2 = \mathbf{f} \end{cases} \tag{6}$$

where $\boldsymbol{\rho}$ is the vector of element-based design variables; $\bar{\boldsymbol{\rho}}$ is the vector of physical densities either computed based on erosion threshold if associated with η_{ero} or on dilation threshold if associated with η_{dil}; $\mathbf{f}$ is the external applied force; $\mathbf{u}_2$ is the field of nodal displacements; v_i is the element volume; V^{**} is the maximal admissible volume for the continuum structure; $\mathbf{k}_i^0$ is the stiffness matrix of the i-th element for the case where $E = 1$. E_{eff} is the modified stiffness, given by (7).

$$E_{\text{eff},i} = \underbrace{(E_{\min} + \tilde{\boldsymbol{\rho}}^{p,\text{ero}}(E_0 - E_{\min}))}_{E_{\text{SIMP}}}(1 + \mathrm{T}_i(\alpha - 1)) \tag{7}$$

with E_{SIMP} given by the modified SIMP model [35,36], in which E stands for the interpolated Young's modulus, $E_{\min}$ being its lower bound – the stiffness of void – and E_0 the stiffness of the base material; p is the penalty factor and α is the influence factor, which physically defines the influence strength. The problem is solved with the MGCG algorithm [37,38], solving iteratively on different grid coarsenesses.

For this formulation, the sensitivities can be calculated with the chain rule. Indeed, the sensitivity of a functional ψ that can represent either the objective or a constraint is given by (8):

$$\frac{\partial \psi}{\partial \boldsymbol{\rho}} = \frac{\partial \psi}{\partial \bar{\boldsymbol{\rho}}} \frac{\partial \bar{\boldsymbol{\rho}}}{\partial \tilde{\boldsymbol{\rho}}} \frac{\partial \tilde{\boldsymbol{\rho}}}{\partial \boldsymbol{\rho}} \tag{8}$$

where $\boldsymbol{\rho}$ is the vector of original design densities; $\tilde{\boldsymbol{\rho}}$ is the vector of the filtered field and $\bar{\boldsymbol{\rho}}$ is the vector of the projected field – dilated for the objective function and eroded for the constraint. The gradients with respect to the projected field are computed using the adjoint method and are given by (9)

$$\frac{\partial \psi}{\partial \bar{\rho}_i} = -\mathbf{u}^{\mathsf{T}} \frac{\partial \mathbf{K}}{\partial \bar{\rho}_i} \mathbf{u} \tag{9}$$

where $\mathbf{u}$ is the displacement field and $\mathbf{K}$ the stiffness matrix. The derivatives $\frac{\partial \bar{\rho}}{\partial \tilde{\rho}}$ and $\frac{\partial \tilde{\rho}}{\partial \rho}$ can be derived from Eqs. (5) and (3), respectively.

2.2. Implementation

2.2.1. Tools

If it was possible on the 2D version of the method to run all the calculation on a personal computer with MATLAB, the increased demand in memory in 3D makes it very difficult. It has then been decided to optimize the MATLAB script by code vectorization *for each operation* for the truss part, and to use parallel computation with PETSc on a cluster for the continuum optimization. For this second step, advantage had been taken of the development of the general framework for topology optimization presented in [37]. Further fundamental function development such as importation of the raster optimal truss or application of the Heaviside filter had been added. These are discussed in the next paragraph. As a matter or comparison, the time needed to solve the 3D cantilever presented at the end of [27] has been reduced by a ratio of 5.

If the work domain is not box-shaped, the geometry must first be defined in a CAD system. STL files were used to import the geometry into MATLAB and PETSc, hence the choice of the modeler was not important. Visualization of the results and unprintability computation were performed with ParaView, a VTK-based open-source, multi-platform data analysis, and visualization application. One of the numerous interests of such a tool is the ease of visualization of the unprintable zones, as illustrated in Fig. 22.

2.2.2. Method

The need to build a robust algorithm that could operate industry-class optimizations justified an extension of the capabilities demonstrated on the 2D domains. Indeed, we need to consider not only box-shaped work domains, but any kind of domains, including convex or non-convex ones. To do so, the workflow presented in Fig. 4 explains the process, and Fig. 5 illustrates the operations on a simple example in 2D, without loss of generality. The diagram is built with the standard notation: the round-angle rectangles are inputs from the user, sharp-angle rectangles are operations performed by the script, and the diamond-like shape is a condition.

In a first stage, we input different elements to the algorithm:

- the coarse and fine grid parameters,
- the loads direction and magnitude,
- the STL files to define the geometries.

As in [27], the coarse grid is constituted of the connecting nodes of the ground structure and the fine grid is the discretized continuum domain. In the case the design domain is defined from an STL file, it is possible to restrain the creation of the coarse grid to the interior of the geometry. If not, some nodes may appear in the voids, as illustrated in Fig. 5(b). It is important to note that some fixed features that are not part of the design domain – not subject to optimization – can also be added. The nodes are then connected to create the ground structure. The connectivity – i.e.

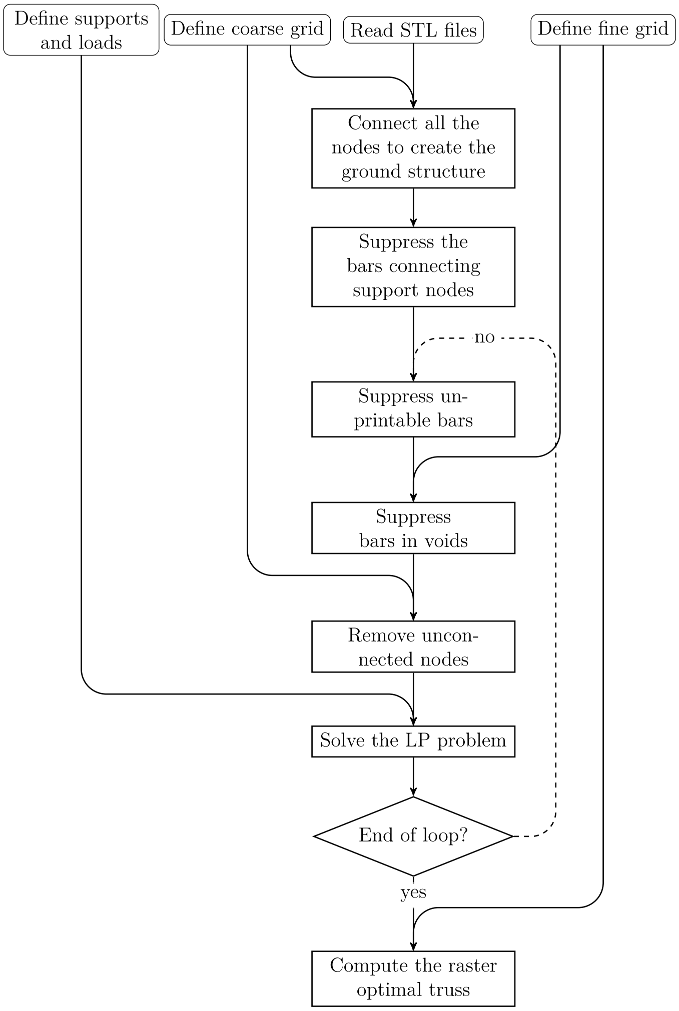

Define supports and loads
Define coarse grid
Read STL files
Define fine grid
Connect all the nodes to create the ground structure
Suppress the bars connecting support nodes
no
Suppress unprintable bars
Suppress bars in voids
Remove unconnected nodes
Solve the LP problem
End of loop?
yes
Compute the raster optimal truss

Fig. 4. Workflow of the first optimization step.

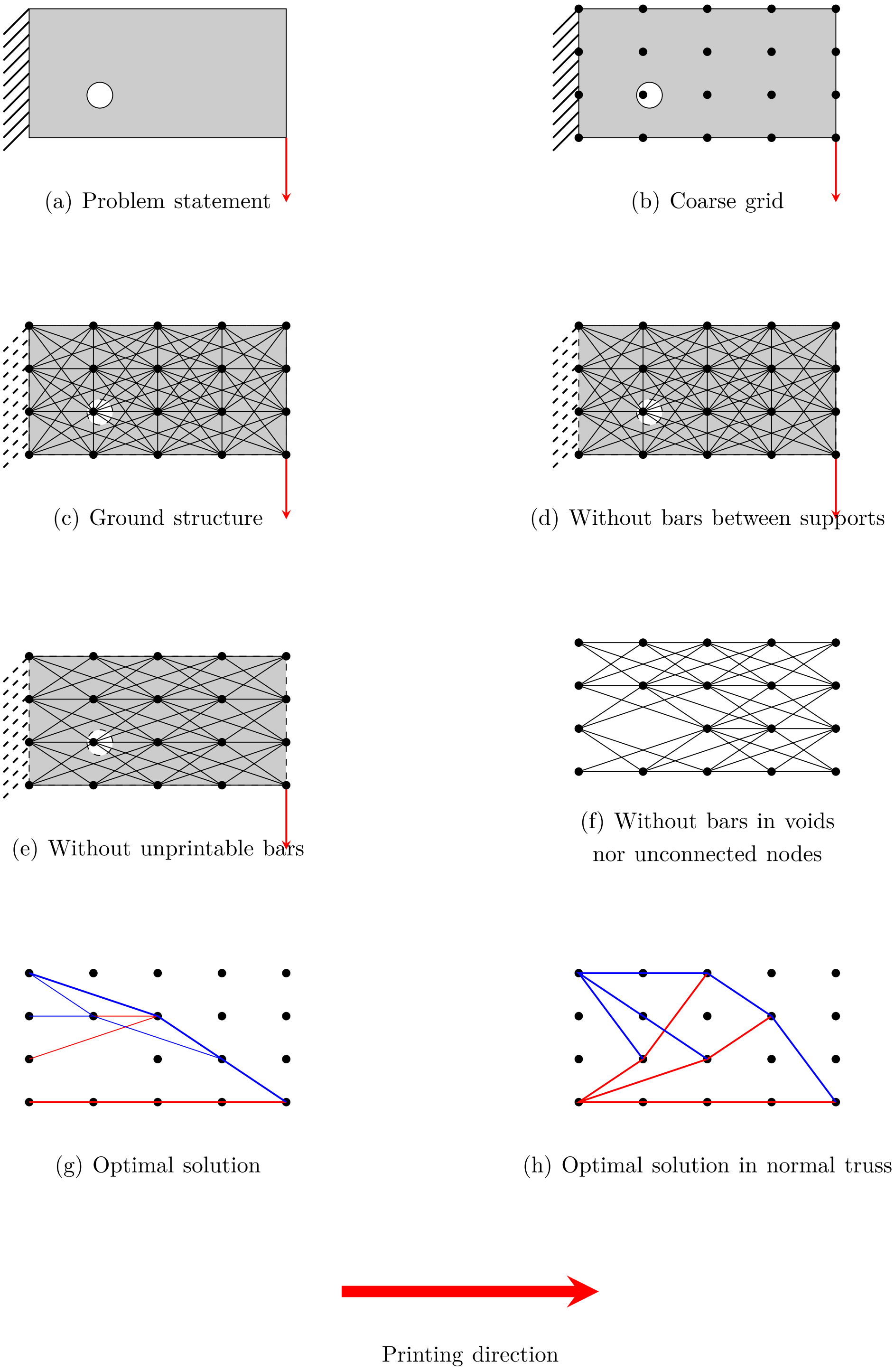
the number of connecting neighboring nodes – can be defined at this point; Fig. 5(c) depicts a connectivity of 2, because each node is not only connected to its closest neighbor, but to all of them within the distance of 2 nodes along each axis. The ground structure at this stage contains the maximum bars possible and must be cleaned. To improve readability, this cleaning process has been split into smaller steps.

In the first cleaning step, we remove the bars connecting support nodes, as we already know that these will not belong to the optimal design, as in Fig. 5(d). As all the bars of the ground structure will be considered in the LP formulation, this operation can greatly alleviate the computational burden, especially when there are many support nodes.

In the second cleaning step, the unprintable bars – the ones that present an exceeding overhang angle – are removed, as in Fig. 5(e). This step and the previous one being independent, they can then be inverted.

In the last cleaning step, the bars that are at least partly located in voids must be removed. This can be performed by computing the distance between each element of the fine grid located in the void to each bar. The finer the mesh, the more precise the cleaning and the more time it takes. This is the reason why this step must be the last one of the cleaning process. The ground structure at the end of this process is represented in Fig. 5(f). In some cases, particularly when working with real work domains, some nodes are completely disconnected from the ground structure after the cleaning process. This is the case in the last example given: due to the opening intended for screw assembling, no bar can reach the support nodes without entering the opening void. Then the initial supports cannot be connected to the ground structure. In this kind of situation, one needs to select neighboring nodes and give them a supporting role. That can be performed easily by using the natural vertices of the STL files.

Disconnected nodes are a problem, as they are subject to rigid body movement and can hence make the stiffness matrix singular. In consequence, they need to be tracked and removed. To increase readability and save memory, we chose to renumber the remaining nodes after this operation. The problem is then ready for optimization. In Fig. 5, the difference between the optimal truss obtained by this method (g) or by the usual optimization algorithm (h) is illustrated, the blue



(a) Problem statement

(b) Coarse grid

(c) Ground structure

(d) Without bars between supports

(e) Without unprintable bars

(f) Without bars in voids nor unconnected nodes

(g) Optimal solution

(h) Optimal solution in normal truss

Fig. 5. Step illustrations.

color depicting tension in the bar, and the red one compression. It can be seen that in (h) the solution comes closer to the Mitchell analytical designs, with an arch that transmits the load to the supports. In (g), there cannot be this arch because, as unprintable, it does not appear in the ground structure. The force must then follow another path, less efficient, imposed by the restricted possibilities given, through an arch-like shape with a remote center. Another important difference between these results lies in the supporting area, on the left. Because of the impossibility to use a bar in the region of the hole, the supporting system has to move upward, which is less efficient.

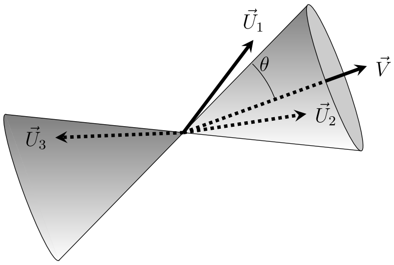


Fig. 6. Solid angle covering.

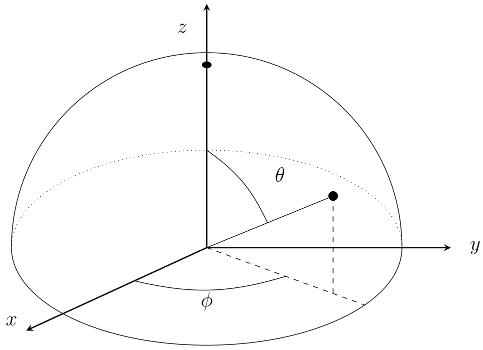


Fig. 7. Solid angle notation.

A raster representation of that resulting truss must be built on the continuum, the "fine" grid, and imported to the parallel computer. The continuum optimization is performed in this latter tool. As mentioned before, the reference [37] has been used as a base. This code has been enriched by three main functions:

1. *the truss importation function*: as an input of the continuum optimization, it is necessary to import the optimal truss into the algorithm with a parallel construction;
2. *the robust formulation* of the Heaviside projection;
3. the extension enabling to consider several load cases.

In order to obtain smooth designs, a pad – space between the boundary of the design domain and the boundary of the work domain – is systematically added. This pad length is equal to the filter radius [32,39].

2.2.3. Optimal printing direction search algorithm

In the case one would like to explore the solid angle to find the best printing direction possible based on the truss representation, a loop must be added to the process. As the workflow is the same from the beginning of the algorithm until the end of the first cleaning step, that loop must only include the operations from the unprintable bar removal stage until the end of the optimization of the discrete problem.

It is important to note that due to the non-directionality of the bar, only π steradians need to be considered, as illustrated in Fig. 6. Indeed, if $\vec{V}$ is the printing direction, the overhang angle is given by θ and the $\vec{U}_i$ are the direction vectors for bars 1 to 3, not only $\vec{U}_2$ and $\vec{U}_3$ are printable directions, but also $-\vec{U}_2$ and $-\vec{U}_3$. Conversely, bars directed by $\vec{U}_1$ or $-\vec{U}_1$ would not be printable.

If the main axes are named x, y and z, any direction can be given by the variables θ and ϕ, as illustrated in Fig. 7. It is enough to have both angles vary between 0 and π radians to cover all the possible directions. Furthermore, the relations from spherical to Cartesian coordinates are given by (10):

$$\begin{cases} x = \sin(\theta)\cos(\phi) \\ y = \sin(\theta)\sin(\phi) \\ z = \cos(\theta) \end{cases} \tag{10}$$

Finally, the easiest manner to implement the search algorithm is to define the angle increments by dividing the 0 to π segment for each of the angles. If one wants to calculate the optimal truss for 13 directions regularly distributed for θ and 15 for ϕ – meaning 195 optimizations – in order to obtain a result similar to the one in Fig. 14, Algorithm 1 can be used. If ones only needs to compute the best printing direction, it is enough to initiate the variables `bestCompliance` and `bestPrintDirection`, check at each iteration if the new compliance is better than `bestCompliance`, and update `bestPrintDirection` accordingly. Such a loop has been implemented in the numerical examples and its output has been presented.

Algorithm 1 Search for the best printing direction.

```
bestCompliance $\leftarrow \infty$
divTheta $\leftarrow 13$
incrTheta $\leftarrow \pi$ / divTheta
divPhi $\leftarrow 15$
incrPhi $\leftarrow \pi$ / divPhi
compliances($i$, $j$) $\leftarrow 0$   $\forall i = 1, \ldots,$ divTheta,   $\forall j = 1, \ldots,$ divPhi
posTableTheta $\leftarrow 0$
for $\theta$ from incrTheta to $\pi$ incremented by incrTheta do
  posTablePhi $\leftarrow 0$
  posTableTheta $\leftarrow$ posTableTheta $+1$
  for $\phi$ from incrPhi to $\pi$ incremented by incrPhi do
    posTablePhi $\leftarrow$ posTablePhi $+1$
    printDirection $\leftarrow [\sin(\phi) \cdot \cos(\theta); \sin(\phi) \cdot \sin(\theta); \cos(\phi)]$
    Solve the optimization problem, refer to Eq. (6)
    compliances(posTableTheta, posTablePhi) $\leftarrow$ compliance for this direction
    if compliance < bestCompliance then
      bestCompliance $\leftarrow$ compliance
      bestPrintDirection $\leftarrow$ printDirection
    end if
  end for
end for
return bestCompliance
```

2.2.4. *Performance computation*

Figure 4 presents the workflow to optimize any 3D design according to the method presented. Once the optimized design is obtained, one wants to compare it to a reference solution, hence the need to compute performance – compliance in the case of the formulation presented – and printability. Regarding the *performance objective*, the virtual increase of Young's modulus to influence the material repartition behaves as if the design was stiffer. Hence, it is mandatory to recalculate the compliance of the influenced model on the same basis as the reference one in order to compare them. As a reference, we use the same script with $\alpha = 1$, meaning there is no influence of the optimal truss. As the method introduces supplementary constraints, we expect the compliance of the reference to be smaller than the one of the influenced structure. Then regarding *relative performance drop*, we will use the relation given by (11):

$$\mathrm{PD} = \frac{C_{\mathrm{infl}} - C_{\mathrm{ref}}}{C_{\mathrm{ref}}} \cdot 100 \tag{11}$$

where C_{infl} and C_{ref} are respectively the nominal compliances of the influenced design and of the reference. This is clear that if the influenced method reaches a better optimum than the reference, PD will be negative.

Regarding *printability* calculation, two options are possible. Even if using the same validation criterion as the one presented in [27] is valid – counting unprintable cells, we preferred computing the value directly in ParaView. The principle is the same, but instead of dividing the number of unprintable elements by the total number of elements, we divided the unprintable surface by the total surface of the part: the isovolume is constituted based on the vector of densities, and the normals to each triangle is computed. A dot product is then operated between these normals and the printing direction, and only the elements whose normals exceed the threshold are sorted out, the threshold depending on the maximum overhang angle. These elements are then summed and constitute the unprintable surface.

An issue remains in problems defined as box-shaped design domains, as in many cases, plates perpendicular to one of the axes could appear and be counted as large unprintable surfaces. The latter can be parts of the solid regions and, as such, known in advance. In this case, it is possible to limit the threshold so as not to include their normals in the unprintability calculation. In our tool, we include the surfaces that present a normal which normalized dot product with the printing direction produces a result between $1/\sqrt{2}$ – for 45° – and 0.999 instead of 1 – along the printing direction, in order to give a more realistic picture of the situation. The *relative unprintability* (UP) of a part is computed as a comparison to a reference in order to obtain a meaningful value. We will use the relation given by (12):

$$\mathrm{UP} = \frac{\mathrm{UP}_{\mathrm{ref}} - \mathrm{UP}_{\mathrm{infl}}}{\mathrm{UP}_{\mathrm{ref}}} \cdot 100 \tag{12}$$

where $\mathrm{UP}_{\mathrm{infl}}$ and $\mathrm{UP}_{\mathrm{ref}}$ are respectively the nominal unprintable surface of the influenced design and the one of the reference. This is clear that if the influenced method is less printable than the reference, UP will be negative. In consequence, the first measurement of the performance of the presented method consists in obtaining both UP and PD positive.

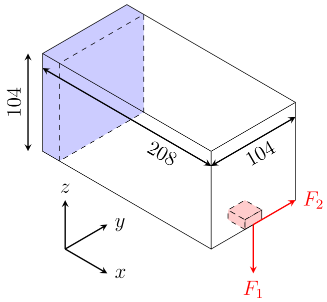


Fig. 8. Convex problem statement.

3. Numerical examples

In all the examples presented and if not stated otherwise, the following values were used:

1) for the density filter and the Heaviside projection
   (a) the radius $r = 4$, in Eq. (3);
   (b) $\eta_{\text{ero}} = 0.6$ and $\eta_{\text{dil}} = 0.4$, in Eqs. (5), (6) and (7);
   (c) β, in Eq. (5), follows a continuation scheme. Its value begins at 1 and is doubled after 25 iterations or if the change in the design is not significant enough, until it reaches 8;
2) the influence parameter $\alpha = 5$, in Eq. (7);
3) the overhang angle is 45°, in the construction of the coarse grid;
4) the continuum simulation is limited to 200 iterations.

3.1. Convex design domain

The design domain for the convex example is a box, like in most academic examples. In Fig. 8, the dimensions as well as the position of the support region (in blue) and the load region (in red) are illustrated. Two load cases are used, the first represented by F_1 and the second by F_2, their magnitude being chosen to unit value. In a situation of several load cases, we want that the *unique* output design would be able to stand each load case *separately*. Then both are considered and the compliance in each load case – depending on the external applied force – are computed separately. The objective function is then constituted by a weighted compliance. In our examples, each load case has equal weight.

For ease and precision, the x, y and z axes are split in respectively 208, 104 and 104 elements of the fine grid; 9, 5 and 5 elements of the coarse grid when the x-axis constitute the printing direction and 7, 4 and 3 elements when the z-axis plays that role. The choice of these numbers is directly linked to the use of MGCG for the continuum problem, which is not the case for the discrete problem. In order to maximize the chances of finding a truss matching the constraints and that contains a limited number of bars, it is useful to ensure that each node can at least connect with its direct neighbors on the face on an element. For example, in this case – printing direction on the x axis with an overhang angle of 45°, the length of the element on x is greater than the ones on y and z. Choosing 10 elements on x might cause rounding errors and suppress this bar; consequently, 9 elements are used in this direction.

The volume fraction V^* is 10%, meaning that we authorize only 10% of the continuum elements to have a density of 1, while the other ones must be completely empty. The optimized design of reference is presented in Fig. 9, the ones resulting from influenced optimization in Figs. 10 and 11. For ease of understanding the method outputs, the rastered optimal truss is represented along with the solid volumes, which are in this example the baseplate at $x = 0$ and the loading zone. A 3D representation has also been added. Using the parameters mentioned hereabove, the results are centralized in Table 1.

It can be noted that in both cases, PD and UP are of the same order of magnitude, meaning that the performance drop in the new design is compensated by a similar amount in printability improvement. Even if only the final user can determine the trade-off he/she wishes between printability or compliance, the nominal values indicate here that when printing along the x, the printability is higher in all the designs compared to z. The degradation of the printability from the truss – completely printable – to the influenced designs can be explained. The designs obtained in Fig. 10 and Fig. 11 show that some beams are privileged by the continuum optimization as they can hold the load better than some truss bars that disappeared. These beams are not printable and hence absent from the truss. Their width is significant, creating large unprintable surfaces.

Finally, it is worth noting that in the case of a convex domain, it is unnecessary to complete the last cleaning step. Indeed, it is known from the beginning that all the nodes that are connected to the ground structure are located inside the design domain. By definition of convexity, all the bars of the ground structure are also completely contained in the domain.

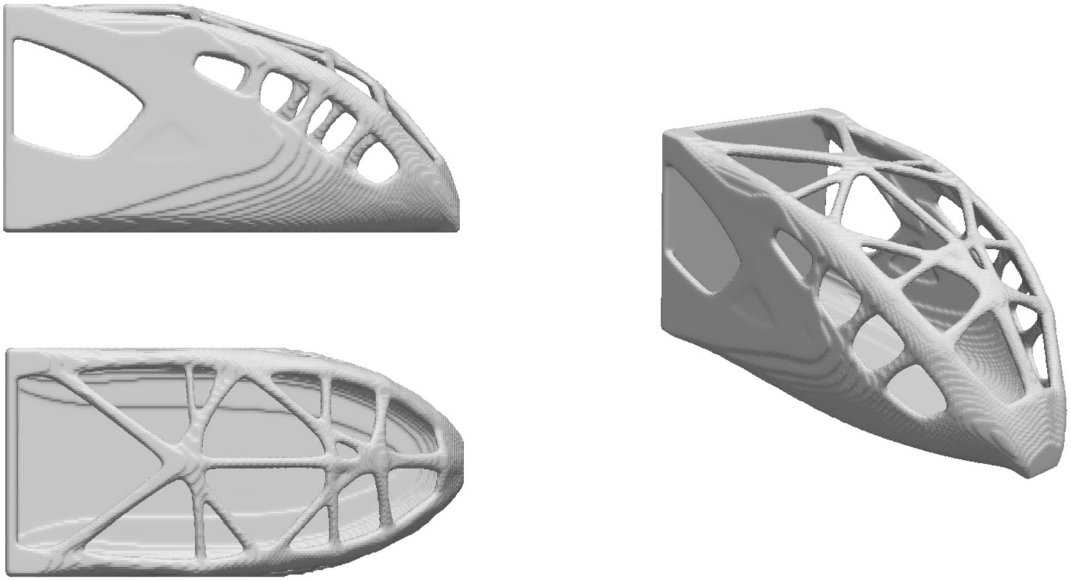


Fig. 9. Reference design from y-axis (upper), z-axis (lower) and 3D (right) views.

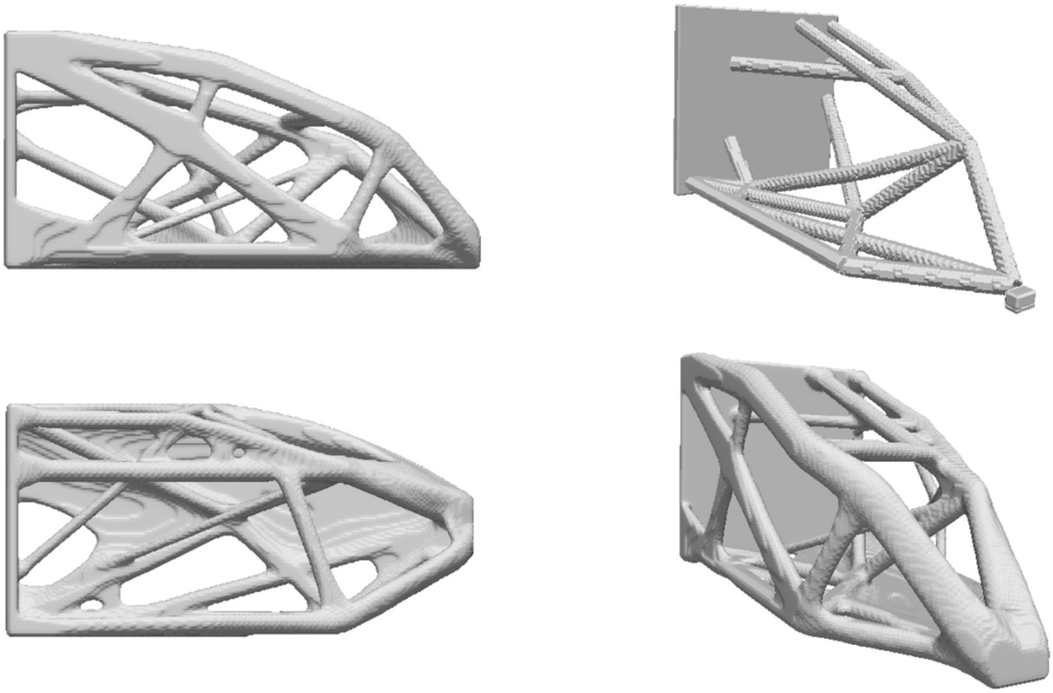


Fig. 10. Influenced designs from y-axis (upper left), z-axis (lower left) and 3D (lower right) views and the rastered truss when printed along the x-axis.

3.2. Non-convex design domain

The design domain for the non-convex example is U-shaped, a box from which a slot has been removed in the middle of one of its faces. In Fig. 12, the dimensions as well as the position of the support region (in blue) and the load region (in red) are illustrated. Two load cases are used, the first represented by F_1 and the second by F_2.

The best printing direction search algorithm has been implemented on this example, when the x, y and z axes were split in respectively 104, 104 and 208 elements of the fine grid and 3, 5 and 9 elements of the coarse grid. Considering that the overhang constraint is only applied in the first stage, this algorithm must be used when solving Eq. (2). The results may obviously be different according to the parameters chosen. To cover the required π sr (steradians), both ϕ and θ were incremented by 12° at each loop iteration – a fifteenth of π radians. The result is presented in Fig. 13. It can be clearly seen

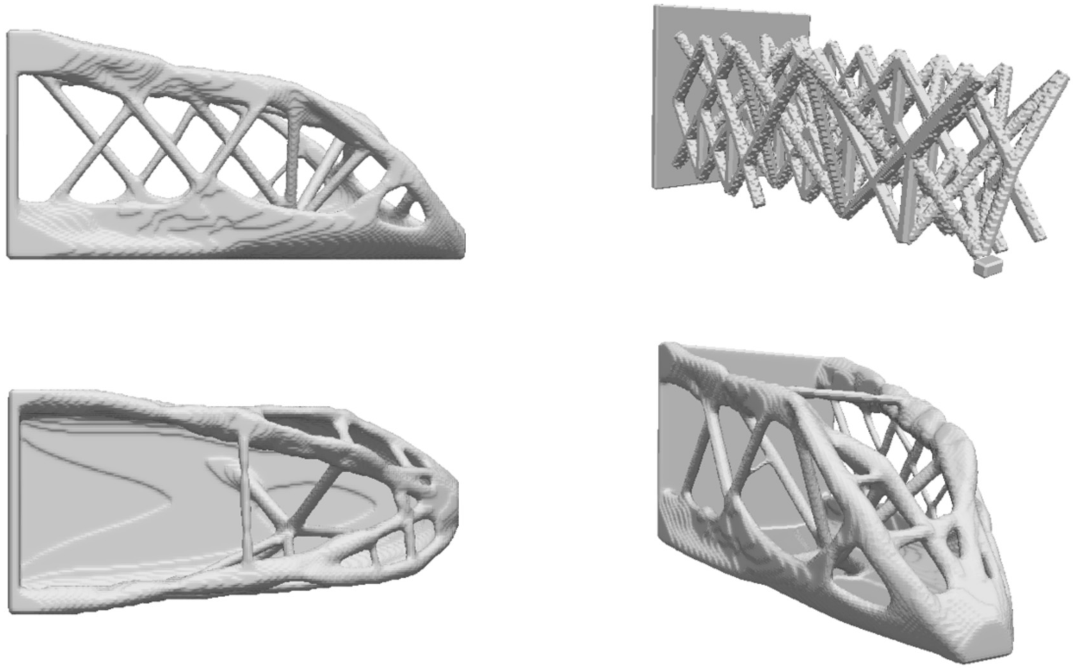


Fig. 11. Influenced designs from y-axis (upper left), z-axis (lower left) and 3D (lower right) views and the rastered truss when printed along the z-axis.

Table 1
Comparison of the influenced and the reference design on the convex domain.

	Printed along x	Printed along z
PD	27.2%	12.8%
Nom. unprint. reference design	2577	5163
Nom. unprint. influenced design	1601	4750
UP	37.87%	8%

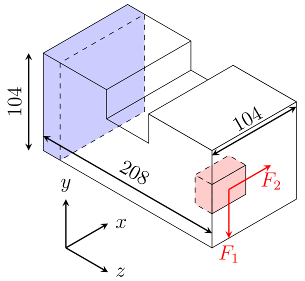


Fig. 12. Non-convex problem statement.

that when θ is close to 0 (or π), there is almost no influence of ϕ, as in the case where there is no change in direction, as depicted in Fig. 7.

Due to the loose discretization of the domain, it can be seen in Fig. 13 that no solution can be found for a certain number of printing directions. In our example, 23 simulations did not converge out of 225. This implies that in some cases, a finer resolution of the solid angle is necessary to obtain valid ground structures. To give a better representation of the result, an interpolation has been realized in Fig. 14 to show the global picture if all the simulations had converged. Not surprisingly, the best printable direction for this example is along the z-axis.

In this specific application, because of the geometry of the design domain, the result provided by the algorithm was logical hence expected. One can benefit from the power of such a tool in a number of situations, inter alia if the geometries are complex and there is no clear "long side", or if the cubicle in the machine is not large enough to print the part in any direction. Then, finding the best direction among a certain allowable solid angle is important. The reference as well as the design obtained while printing in the z direction or in the x one are presented. It can be observed that, in

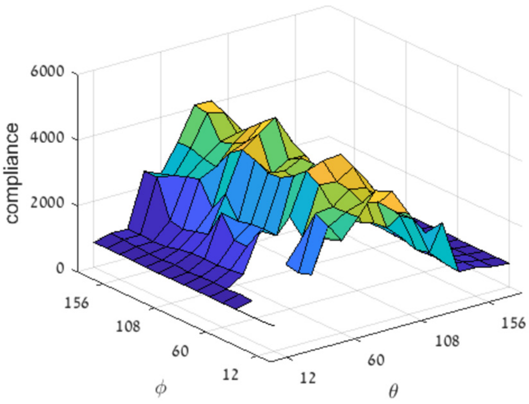


Fig. 13. Initial compliances.

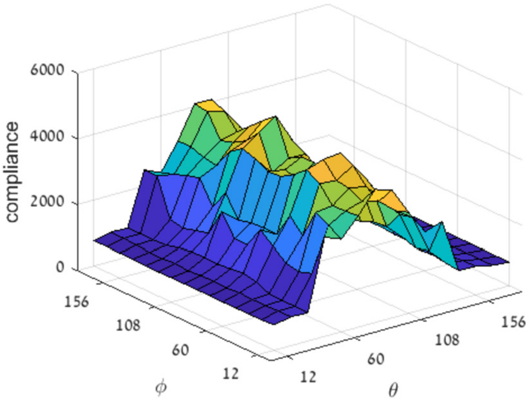


Fig. 14. Compliances with interpolation.

both cases, printability with respect to the reference has been improved. This observation cannot be made from Fig. 14 because compliance is the sole objective function of the discrete optimization problem, as by construction the optimal truss is completely printable. Besides, compliances are better while printed in z than the one printed in x for both the reference and the influenced design. These are presented in Fig. 15 for the reference and Figs. 16 and 17 for the influenced solutions. The volume fraction has been fixed to 8% of the total volume. Table 2 presents the relative performance drop and unprintability calculation of the influenced designs with respect to the reference, as well as their nominal values. It can be noted that, as in the previous case, there is a clear "better choice" as the printabilities along the z, for both designs have better printabilities. Then, either the user will want to privilege the compliance and the reference will be printed, or printability is more important and the influenced design will be chosen.

It is worth noting that the complete scan of the π sr – 225 simulations – took around 3 h 20 min on a personal computer, while one unique continuum simulation on a parallel computer using 4 nodes with 12 threads each took almost 7 h. This confirms the usefulness of this algorithm for preliminary exploration of the printing direction.

3.3. General design domain

The design domain for the general example, whose purpose is to imitate a real industrial case, has been drawn as illustrated in Fig. 18. It is a box of $280 \times 112 \times 112$ mm^3 – the longest dimension being along the x-axis, which should be fixed with 4 screws on a y–z plane. Voids were added in front of their locations as to ensure that no material could prevent the screws from entering their slots. Furthermore, we suppose that another part (for example a tube carrying a fluid) must go through the piece, explaining the supplementary void visible on the x–z plane.

The loads are brought through the blue screw visible on the right side. Two load cases are applied: a force F_1 along the z-axis and another one F_2 oriented in the direction y–z. A thin ring the size of the screw is chosen as a loading region, in order to model the micro-deformations within a small depth under the contact surface, due to the force application.

Finally, a hollow cylinder the diameter of the ring and the depth the distance between the screw and the nut is materialized as a solid region. Small rings surrendering the support screw holes are added to the solid region to ensure that there is enough material to fasten the part after optimization. In this example too, the rastered optimal truss is represented along with the solid volumes. It is interesting to notice the void spaces on the upper right sub-figure in Figs. 20 and 21, and

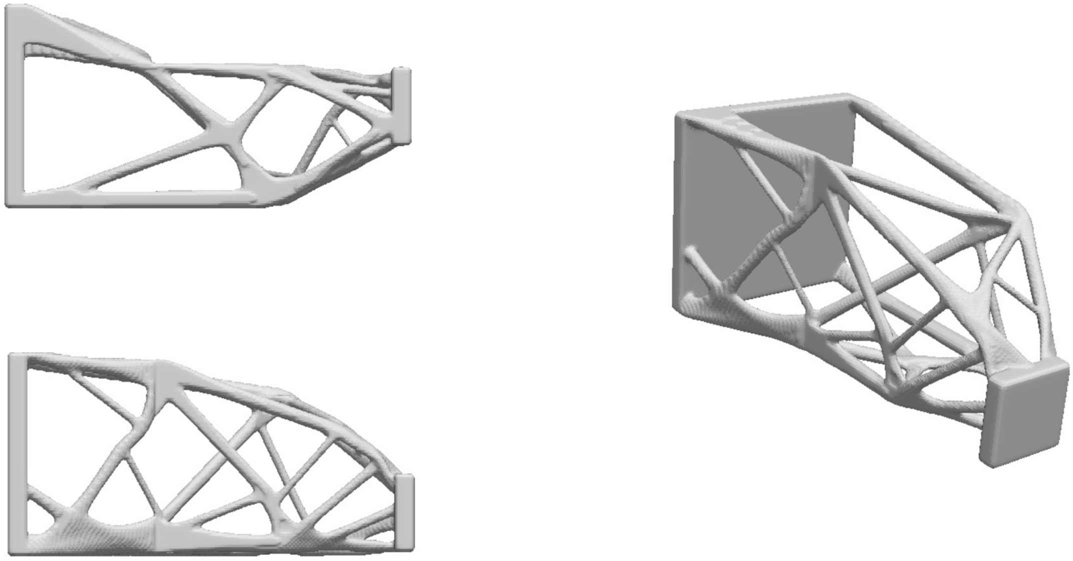


Fig. 15. Reference design from y-axis (upper) and from z-axis (lower) and 3D (right) views.

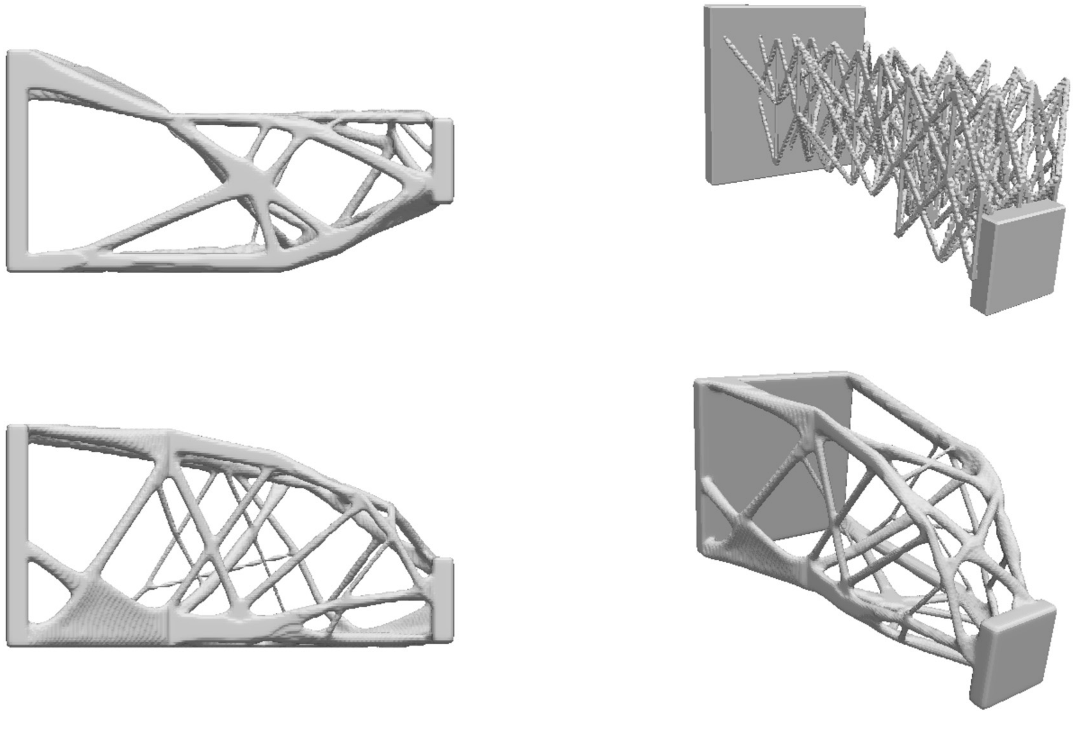


Fig. 16. Influenced designs from y-axis (upper left), z-axis (lower left) and 3D (lower right) views and the rastered truss when printed along the x-axis.

the sharp cut they make between the fastening rings and the truss, which illustrates the difficulty in defining the support and load nodes. In the current case, it has not been possible to define support points only within the solid supporting volume, because due to the screw voids, virtually no bar can effectively connect them to the loading zone. The proper way of treating this problem is to define supplementary supporting nodes *outside the supporting volumes*. Several routines have been developed to select all the natural STL nodes within a certain radius or along a certain direction and to make loading nodes out of them. It is also interesting to note the influence of the geometry itself: if the voids were square, the STL nodes would only have been the square edges, while when filleting their internal side, more nodes are created and an optimal truss is more easily found. This trick is allowed, as when running the continuum optimization, material will naturally enclose the voids and connect the loading rings to the truss. One needs to pay attention to the supplementary cost in material that this covering will induce, to properly define the parameters during truss rasterization.

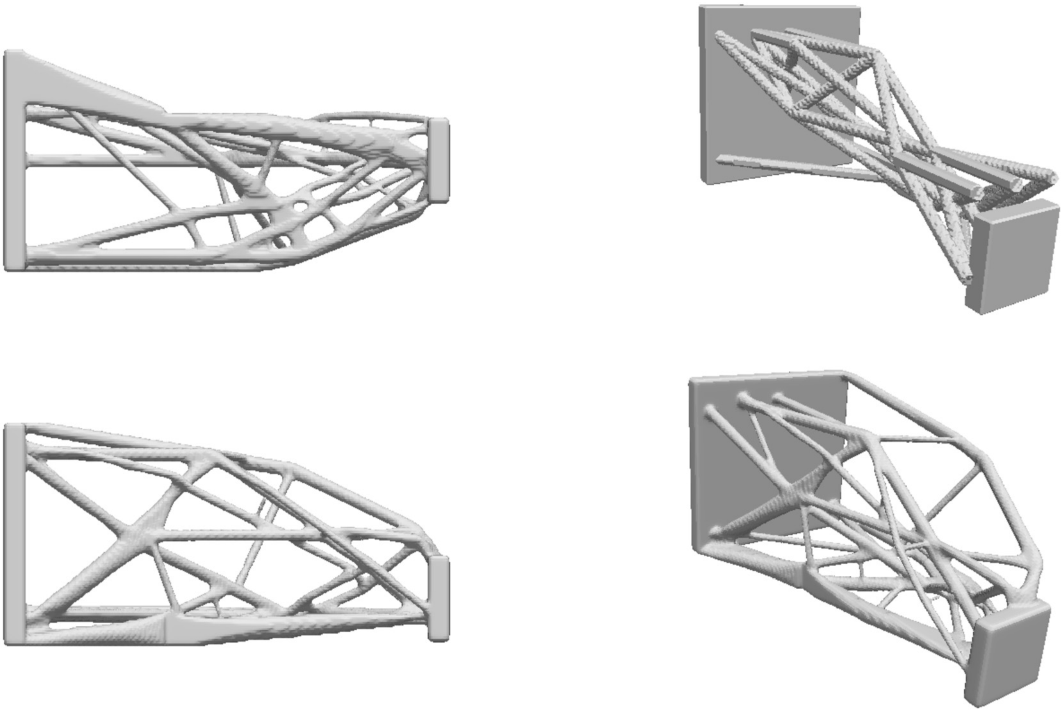


Fig. 17. Influenced designs from y-axis (upper left), z-axis (lower left) and 3D (lower right) views and the rastered truss when printed along the z-axis.

Table 2
Comparison of the influenced and the reference design on the non-convex domain.

	Printed along x	Printed along z
PD	6.35%	32.17%
Nom. unprint. reference design	5268	1886
Nom. unprint. influenced design	4913	1441
UP	6.74%	23.59%

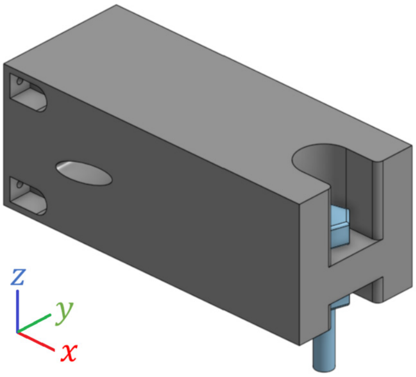


Fig. 18. Engineering problem statement.

Two printing directions were tested: the x-axis and the z-axis. The x, y, and z axes were split in respectively 280, 112, and 112 elements of the fine grid; 6, 4, and 4 elements of the coarse grid when printed along the x-axis and 20, 10, and 10 elements otherwise. Of course, as mentioned, the result obtained for the x (or the z) axis remains valid for the $-x$ (or the $-z$) axis. The results are presented in Table 3 for the values and in Figs. 19, 20 and 21 for the topologies. The same better choice as in the previous cases is noticeable. When printing along the $-x$, the reference design is more printable than any of the options along the $-z$. Hence, without further efforts, we can conclude that it is globally better to print the part along the $-z$. Then, the user will need to opt for the best trade-off regarding compliance and printability.

Table 3
Comparison of the influenced and the reference design on the engineering example.

	Printed along $-x$	Printed along $-z$
PD	20.2%	25.2%
Nom. unprint. reference design	2353	9302
Nom. unprint. influenced design	2240	9047
UP	4.8%	2.74%

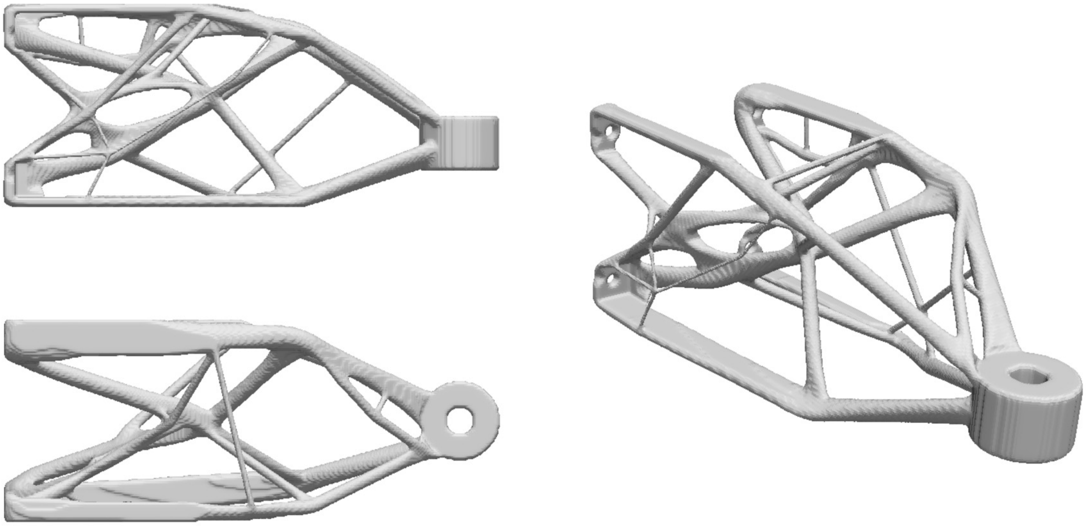


Fig. 19. Reference design from y-axis (upper) and from z-axis (lower) and 3D (right) views.

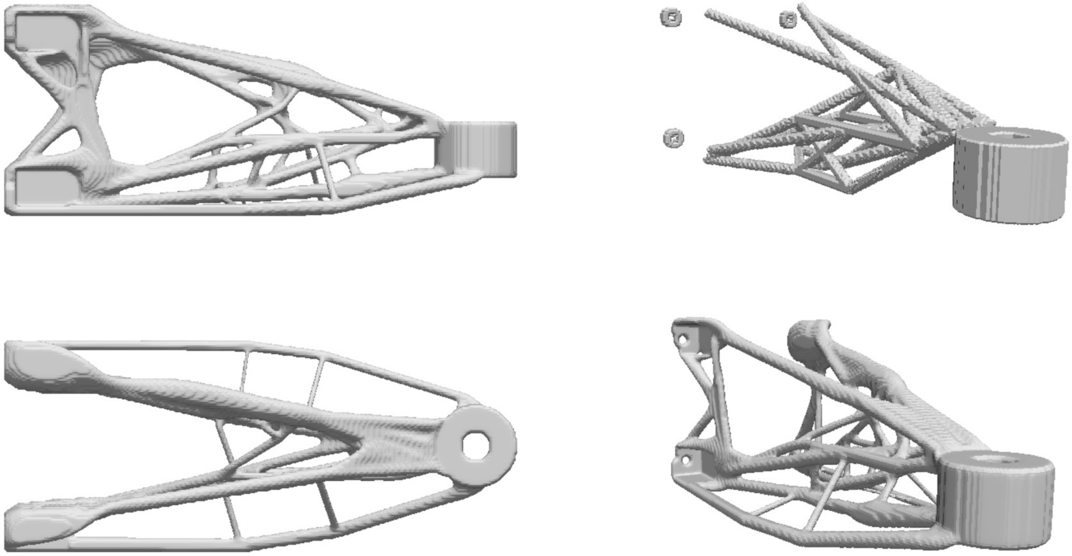


Fig. 20. Influenced designs from y-axis (upper left), z-axis (lower left) and 3D (lower right) views and the rastered truss when printed along the x-axis.

Finally, Fig. 22 depicts the unprintable surfaces while printing on $-x$ and on $-z$.

4. Discussion

The present article extends the method published in [27], whose efficiency has previously been demonstrated on 2D design domains. The idea consists in solving a unique optimization problem, first a discrete manner, then with a continuum formulation, while the result of the first step serves as a skeleton that favors specific zones in the second one. The suggested

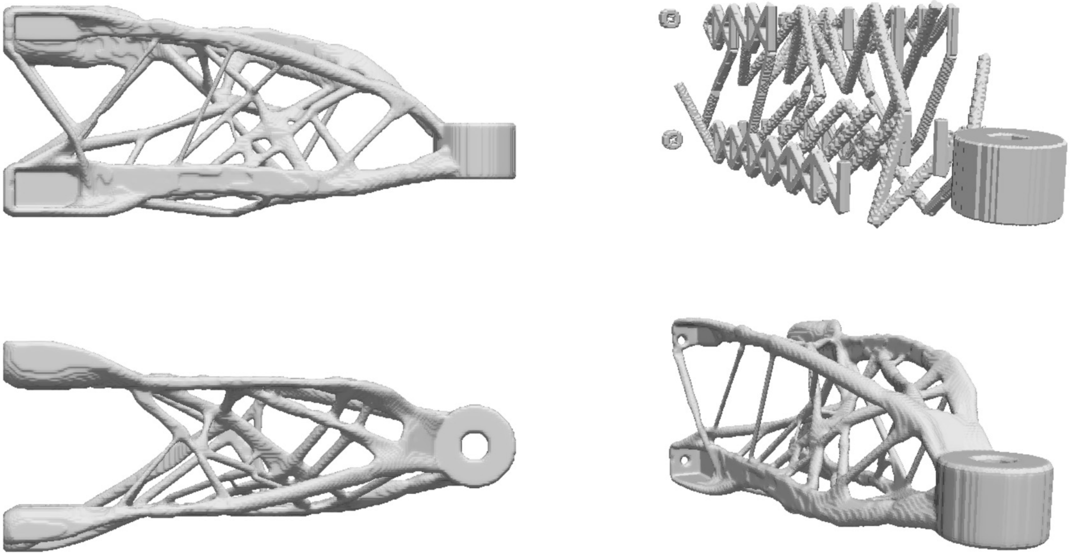


Fig. 21. Influenced designs from y-axis (upper left), z-axis (lower left) and 3D (lower right) views and the rastered truss when printed along the z-axis.

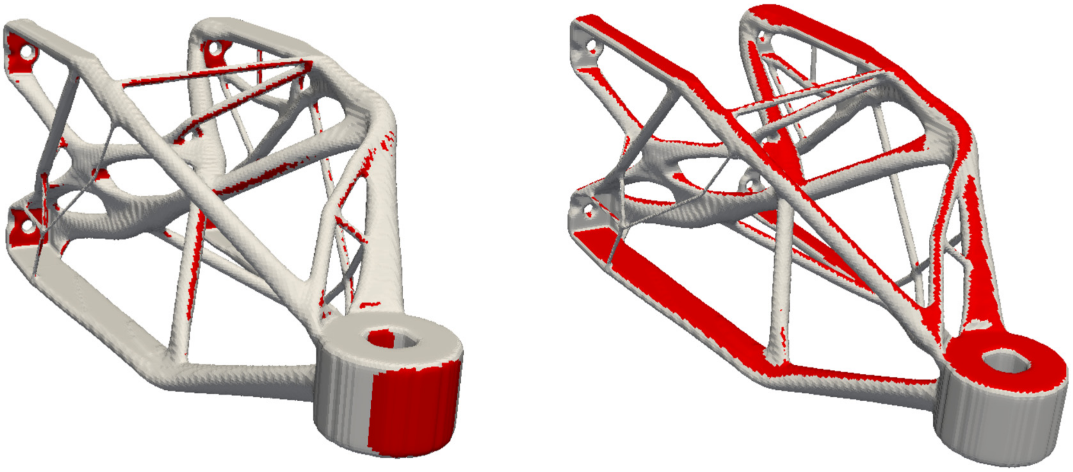


Fig. 22. Unprintable regions for the reference optimal design when printed on $-x$ (left) and on $-z$ (right).

approach accounts for the overhang angle, which is a fundamental limitation in 3-D printing technologies. This restriction appears explicitly in the first optimization step. It has been shown on several examples that this two-act approach can be an efficient manner of reducing the non-printable surfaces while maintaining a high level of performance. Furthermore, the ability of the algorithm for searching the best printable direction has also been demonstrated, taking advantage of the ease of switching the printing direction hence exploring the solid angle. As mentioned for the 2D case, it is generally advantageous to use a grid as coarse as possible to solve the discrete problem, knowing that a more populated truss eventually leads to displaying many bars that, as a suggestion only, would be replaced by more efficient non-printable bars. Finally, the proposed procedure also displays the unprintable surfaces, which can alleviate a post-treatment for efficient support addition. It is important to remember that the unprintability result is an estimation and does not consider the printability over a small distance of unprintable features as well as further steps of the manufacturing process. For example, in the latest domain studied, if the loading hole needs to be located precisely or with a specific surface finishing, it will probably be post-processed by a CNC machine. In this case, there would be no unprintable surface inside the hollow cylinder.

Several leads to continue improving the results exist, inter alia using the skeleton as the initial repartition of material, leading to solutions that are closer to the first optimization step output, or switching from a truss model that cannot transmit moments between its members to a one that can. In this latter case, the output of the first step should be more easily followed by the continuum optimization, leading to a better printability of the final design.

Future works will focus on developing a method that considers the physical root cause of the 3D printing limitation instead of its effect. Indeed, the current research determines designs based on geometrical constraint rather than the real restriction: the overhang angle. However, 1) this angle value changes according to the materials and technologies used, and 2) small overhanged features can be printable.

References

[1] C.W. Hull, Apparatus for production of three-dimensional objects by stereolithography, US Patent 4,575,330, March 11, 1986.
[2] H. Lipson, M. Kurman, Fabricated: The New World of 3D Printing, John Wiley & Sons, 2013.
[3] D. Gu, W. Meiners, K. Wissenbach, R. Poprawe, Laser additive manufacturing of metallic components: materials, processes and mechanisms, Int. Mater. Rev. 57 (3) (2012) 133–164.
[4] W.E. Frazier, Metal additive manufacturing: a review, J. Mater. Eng. Perform. 23 (6) (2014) 1917–1928.
[5] T.T. Le, S.A. Austin, S. Lim, R.A. Buswell, R. Law, A.G. Gibb, T. Thorpe, Hardened properties of high-performance printing concrete, Cem. Concr. Res. 42 (3) (2012) 558–566.
[6] M.P. Bendsøe, O. Sigmund, Topology Optimization: Theory, Methods and Applications, Springer, Berlin, 2003.
[7] M. Bruggi, C. Cinquini, Topology optimization for thermal insulation: an application to building engineering, Eng. Optim. 43 (11) (2011) 1223–1242.
[8] T.E. Bruns, Topology optimization of convection-dominated, steady-state heat transfer problems, Int. J. Heat Mass Transf. 50 (15) (2007) 2859–2873.
[9] J. Alexandersen, N. Aage, C.S. Andreasen, O. Sigmund, Topology optimisation for natural convection problems, Int. J. Numer. Methods Fluids 76 (10) (2014) 699–721.
[10] S. Bose, S. Vahabzadeh, A. Bandyopadhyay, Bone tissue engineering using 3D printing, Mater. Today 16 (12) (2013) 496–504.
[11] I. Gibson, D.W. Rosen, B. Stucker, et al., Additive Manufacturing Technologies, vol. 238, Springer, 2010.
[12] M.P. Bendsøe, N. Kikuchi, Generating optimal topologies in structural design using a homogenization method, Comput. Methods Appl. Mech. Eng. 71 (2) (1988) 197–224.
[13] B. Leutenecker-Twelsiek, C. Klahn, M. Meboldt, Considering part orientation in design for additive manufacturing, Proc. CIRP 50 (2016) 408–413.
[14] A.M. Phatak, S. Pande, Optimum part orientation in rapid prototyping using genetic algorithm, J. Manuf. Syst. 31 (4) (2012) 395–402.
[15] M.P. Zwier, W.W. Wits, Design for additive manufacturing: automated build orientation selection and optimization, Proc. CIRP 55 (2016) 128–133.
[16] D. Thomas, The Development of Design Rules for Selective Laser Melting, PhD thesis, University of Wales, UK, 2009.
[17] D. Brackett, I. Ashcroft, R. Hague, Topology optimization for additive manufacturing, in: Proceedings of the Solid Freeform Fabrication Symposium, Austin, TX, USA, 2011, pp. 348–362.
[18] M. Leary, L. Merli, F. Torti, M. Mazur, M. Brandt, Optimal topology for additive manufacture: a method for enabling additive manufacture of support-free optimal structures, Mater. Des. 63 (2014) 678–690.
[19] A.T. Gaynor, J.K. Guest, Topology optimization for additive manufacturing: considering maximum overhang constraint, in: 15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2014, pp. 16–20.
[20] A.T. Gaynor, J.K. Guest, Topology optimization considering overhang constraints: eliminating sacrificial support material in additive manufacturing through design, Struct. Multidiscip. Optim. 54 (5) (2016) 1157–1172.
[21] M. Langelaar, An additive manufacturing filter for topology optimization of print-ready designs, Struct. Multidiscip. Optim. (2016) 1–13, https://doi.org/10.1007/s00158-016-1522-2.
[22] M. Langelaar, Topology optimization of 3D self-supporting structures for additive manufacturing, Addit. Manuf. (2016), https://doi.org/10.1016/j.addma.2016.06.010.
[23] E. van de Ven, R. Maas, C. Ayas, M. Langelaar, F. van Keulen, Continuous front propagation-based overhang control for topology optimization with additive manufacturing, Struct. Multidiscip. Optim. (2018) 1–17, https://doi.org/10.1007/s00158-017-1880-4.
[24] X. Guo, J. Zhou, W. Zhang, Z. Du, C. Liu, Y. Liu, Self-supporting structure design in additive manufacturing through explicit topology optimization, Comput. Methods Appl. Mech. Eng. 323 (2017) 27–63.
[25] O. Amir, Y. Mass, Topology optimization for staged construction, Struct. Multidiscip. Optim. (2017) 1–16, https://doi.org/10.1016/j.addma.2017.08.001.
[26] G. Allaire, C. Dapogny, R. Estevez, A. Faure, G. Michailidis, Structural optimization under overhang constraints imposed by additive manufacturing technologies, J. Comput. Phys. 351 (2017) 295–328.
[27] Y. Mass, O. Amir, Topology optimization for additive manufacturing: accounting for overhang limitations using a virtual skeleton, Addit. Manuf. 18 (2017) 58–73.
[28] A.M. Mirzendehdel, K. Suresh, Support structure constrained topology optimization for additive manufacturing, Comput. Aided Des. 81 (2016) 1–13.
[29] X. Qian, Undercut and overhang angle control in topology optimization: a density gradient based integral approach, Int. J. Numer. Methods Eng. (2017), https://doi.org/10.1002/nme.5461.
[30] W. Hemp, Optimum Structures, Oxford University Press, 1973.
[31] F. Wang, B.S. Lazarov, O. Sigmund, On projection methods, convergence and robust formulations in topology optimization, Struct. Multidiscip. Optim. 43 (6) (2011) 767–784.
[32] B.S. Lazarov, F. Wang, O. Sigmund, Length scale and manufacturability in density-based topology optimization, Arch. Appl. Mech. 86 (1) (2016) 189–218.
[33] T.E. Bruns, D.A. Tortorelli, Topology optimization of non-linear elastic structures and compliant mechanisms, Comput. Methods Appl. Mech. Eng. 190 (26) (2001) 3443–3459.
[34] B. Bourdin, Filters in topology optimization, Int. J. Numer. Methods Eng. 50 (9) (2001) 2143–2158.
[35] M.P. Bendsøe, Optimal shape design as a material distribution problem, Struct. Optim. 1 (4) (1989) 193–202.
[36] O. Sigmund, S. Torquato, Design of materials with extreme thermal expansion using a three-phase topology optimization method, J. Mech. Phys. Solids 45 (6) (1997) 1037–1067.
[37] N. Aage, B.S. Lazarov, Parallel framework for topology optimization using the method of moving asymptotes, Struct. Multidiscip. Optim. 47 (4) (2013) 493–505.
[38] O. Amir, N. Aage, B.S. Lazarov, On multigrid-cg for efficient topology optimization, Struct. Multidiscip. Optim. 49 (5) (2014) 815–829.
[39] A. Clausen, E. Andreassen, On filter boundary conditions in topology optimization, Struct. Multidiscip. Optim. 56 (5) (2017) 1147–1155.